2. Image Enhancement

The aim of image enhancement is to improve the interpretability or perception of information in images for human viewers, or to provide `better' input for other automated image processing techniques. Image enhancement techniques can be divided into two broad categories:

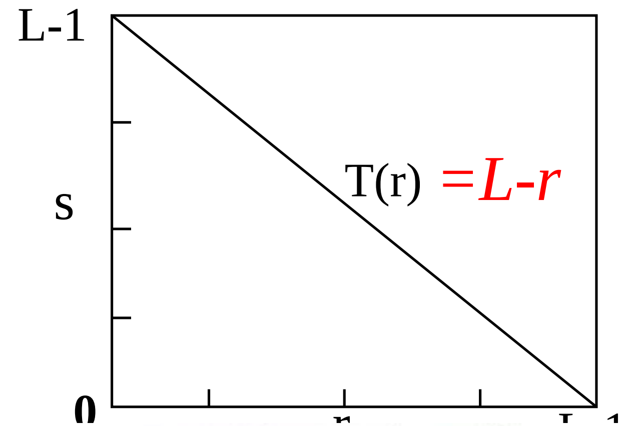
- Spatial domain methods, which operate directly on pixels, and
- Frequency domain methods, which operate on the Fourier transform of an image.

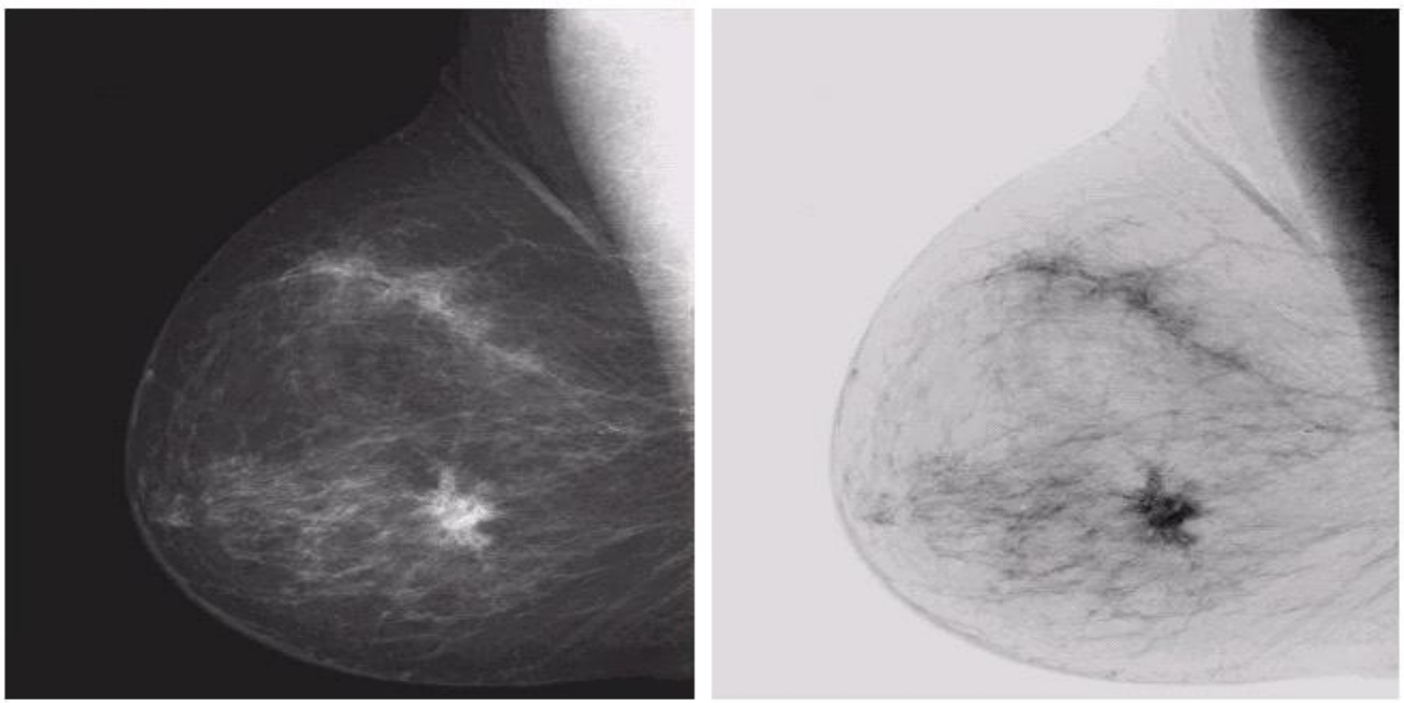
• Chapter 3 (Image Enhancement in the Spatial Domain)

• Section 6.3 (Pseudocolor Image Processing)

Gray Level Transformations: Negative

- **Image Negative:** reverse the brightness from "black" to "white". Useful in displaying medical images.

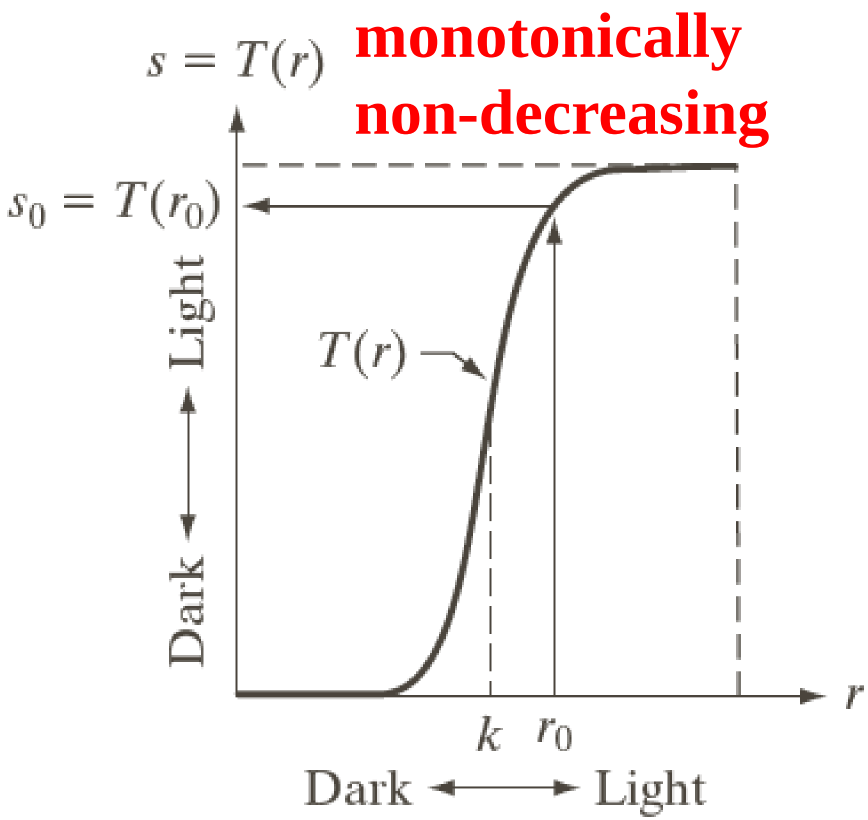


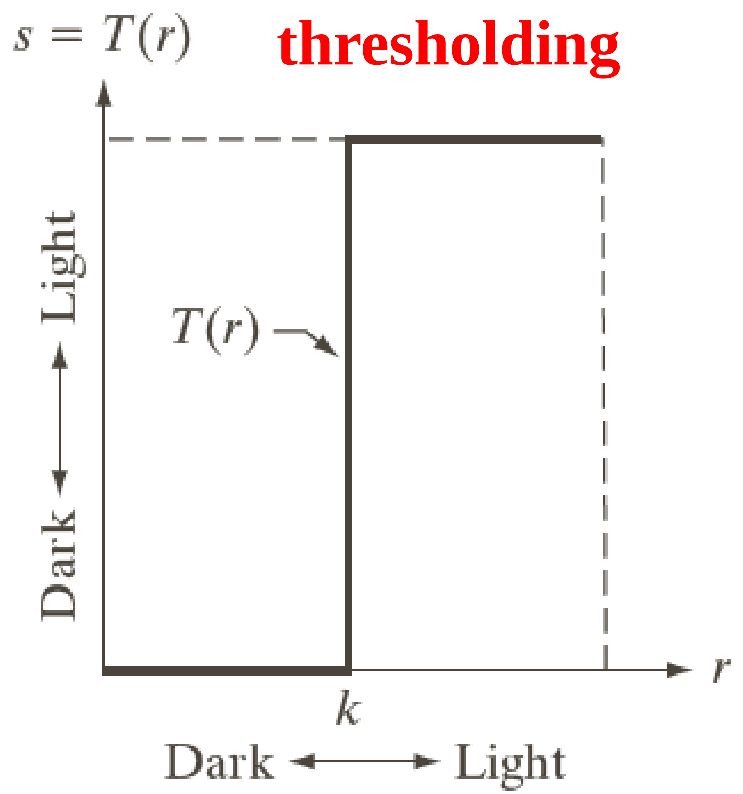


a b

FIGURE 3.4
(a) Original digital mammogram.
(b) Negative image obtained using the negative transformation in Eq. (3.2-1).
(Courtesy of G.E. Medical Systems.)

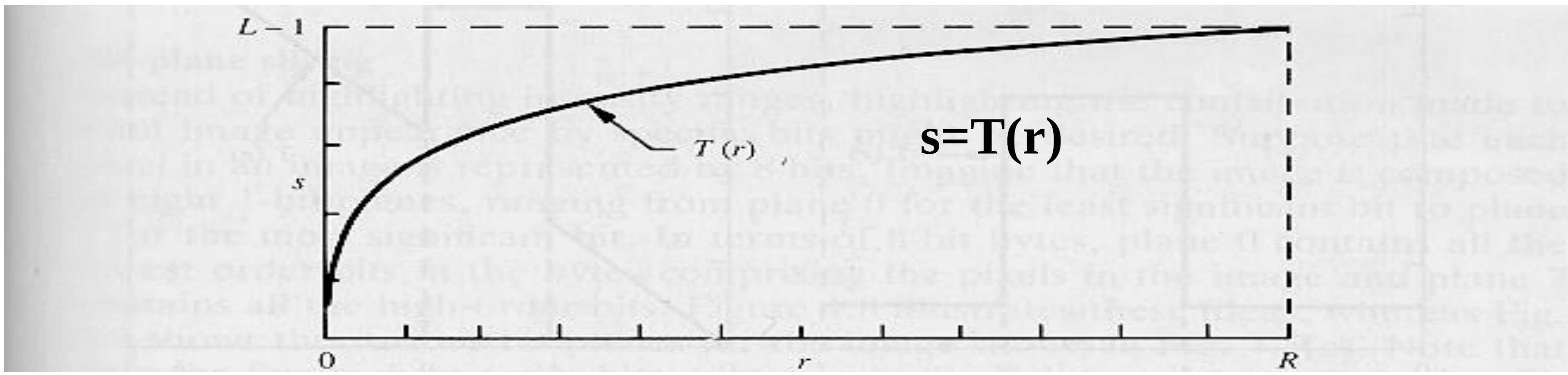
Gray Level Transformation

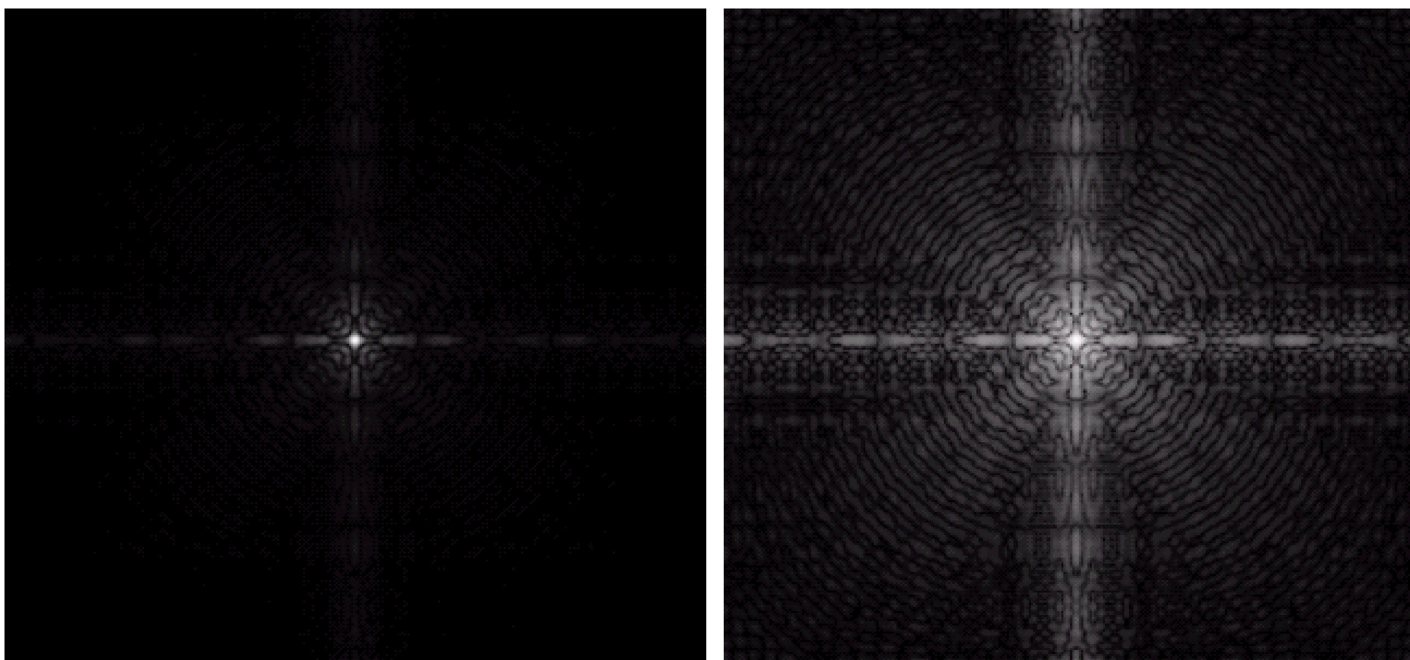




Gray Level Transformations: Dynamic Range Compression

- Processing images exceeding the display capability **s = c log(1+ |r|)**



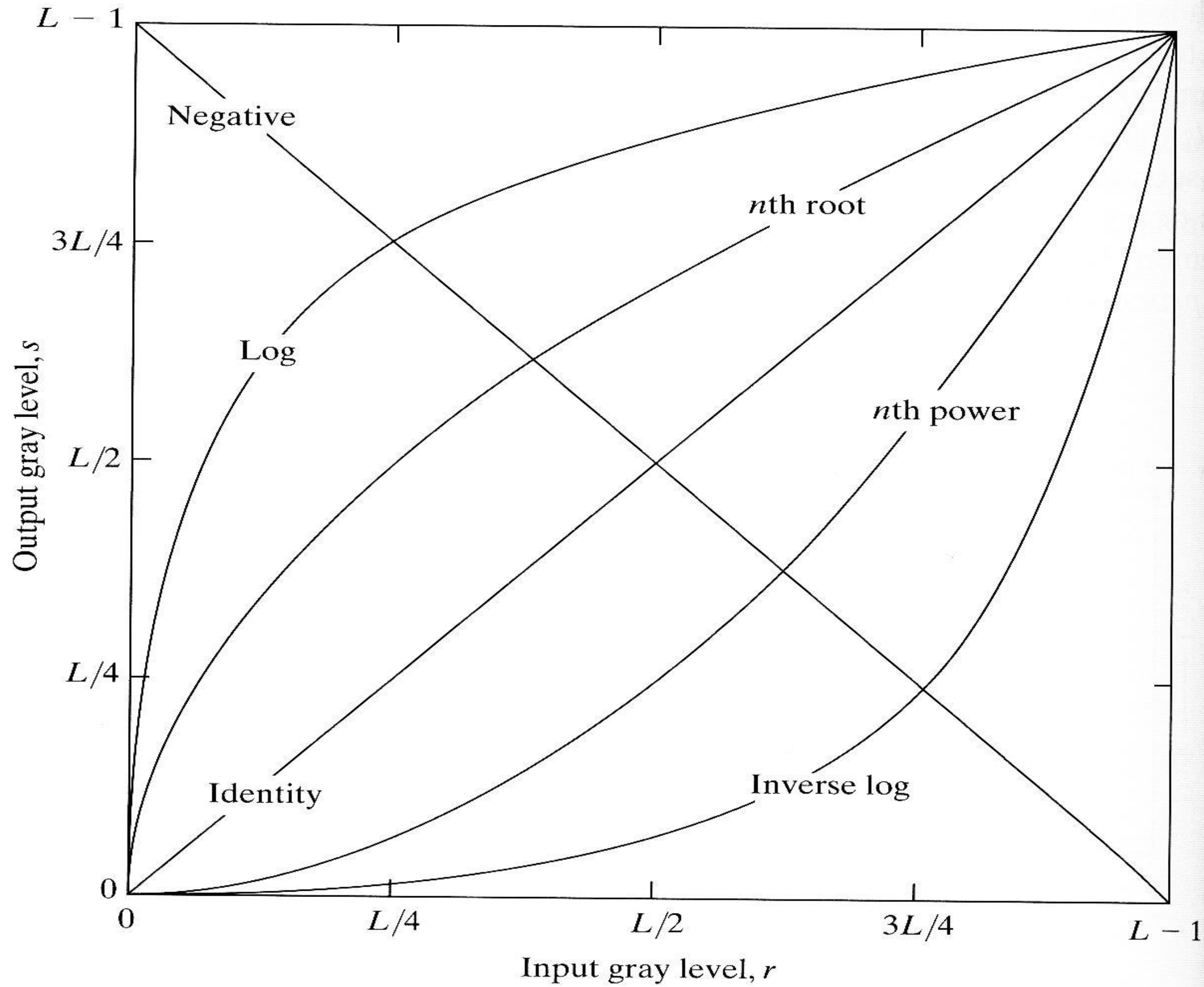


a b

FIGURE 3.5
(a) Fourier spectrum.
(b) Result of applying the log transformation given in Eq. (3.2-2) with $c = 1$.

Image Enhancement by Gray Level Transformations

FIGURE 3.3 Some basic gray-level transformation functions used for image enhancement.



Gray Level Transformations: Power-Law Transforms

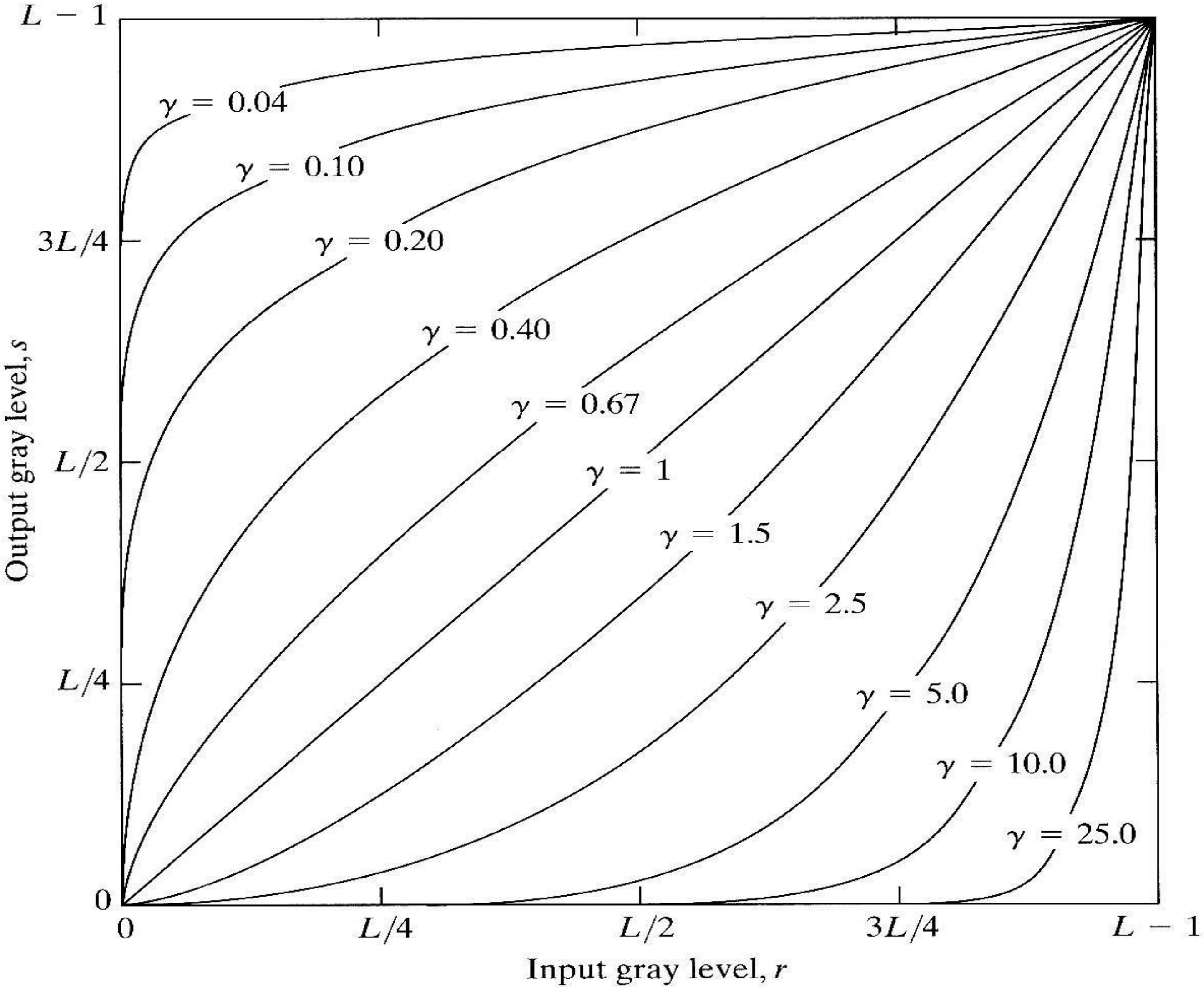


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases).

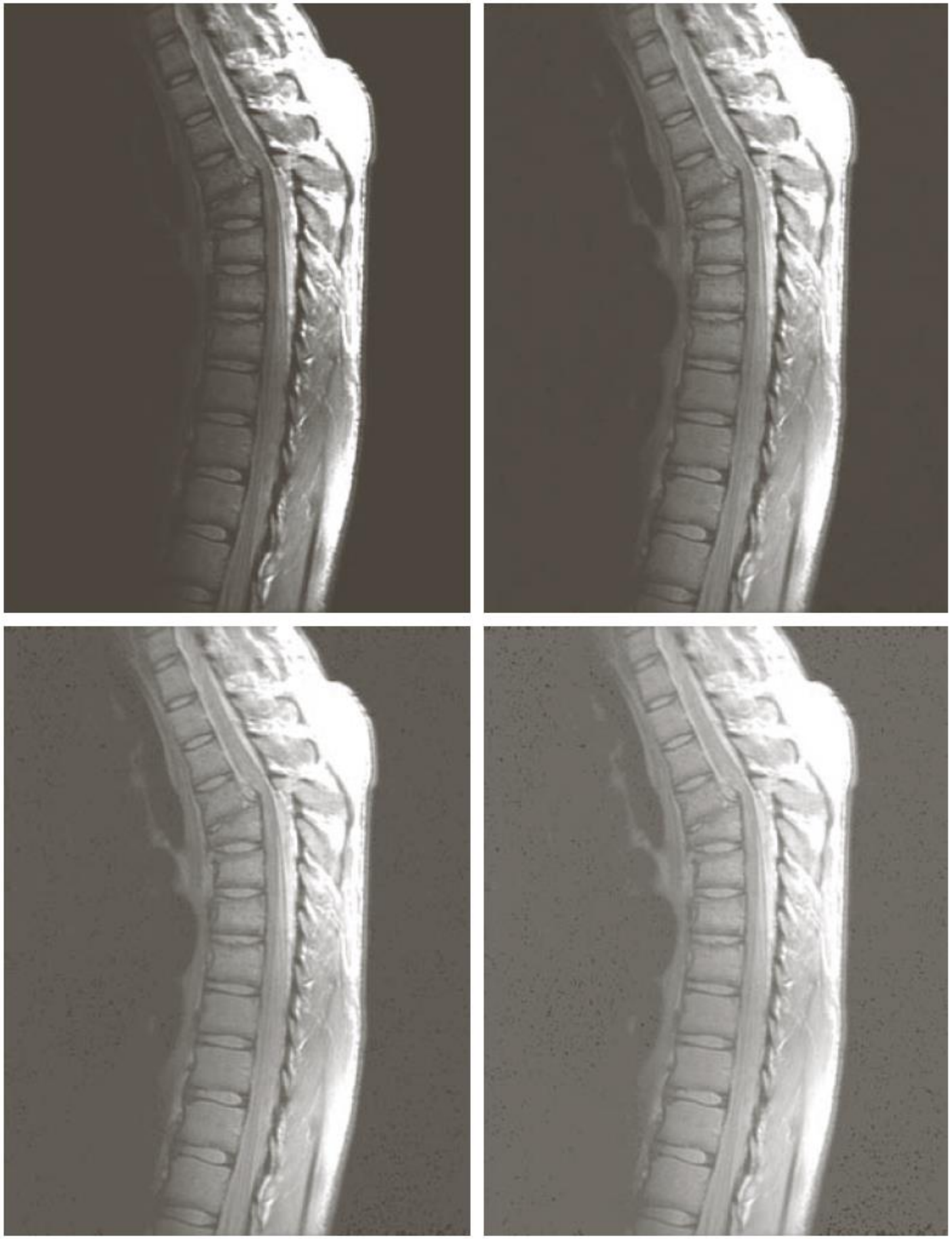
$$s = T(r)$$

$$= cr^{\gamma}$$

Gray Level Transformations: Power-Law Transforms

a b
c d

FIGURE 3.9
(a) Aerial image. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 3.0, 4.0,$ and 5.0, respectively. (Original image for this example courtesy of NASA.)



a	b
c	d

FIGURE 3.8
(a) Magnetic resonance image (MRI) of a fractured human spine.
(b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 0.6, 0.4$, and 0.3, respectively. (Original image courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

Gamma Correction on Monitors

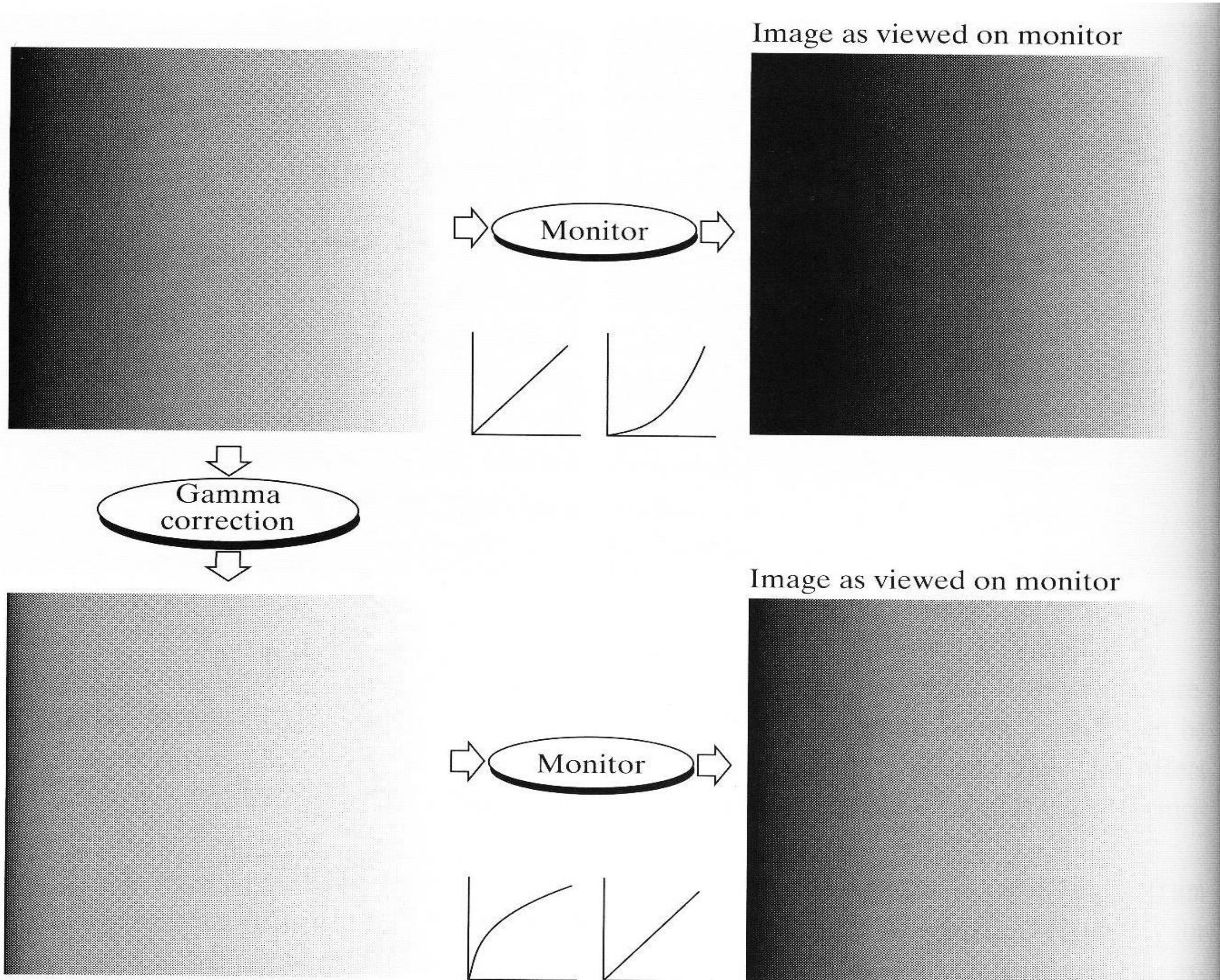


FIGURE 3.7
(a) Linear-wedge gray-scale image.
(b) Response of monitor to linear wedge.
(c) Gamma-corrected wedge.
(d) Output of monitor.

$$\boxed{s = r^{0.4}}$$

Tonal Corrections for flat, light, and dark RGB color images

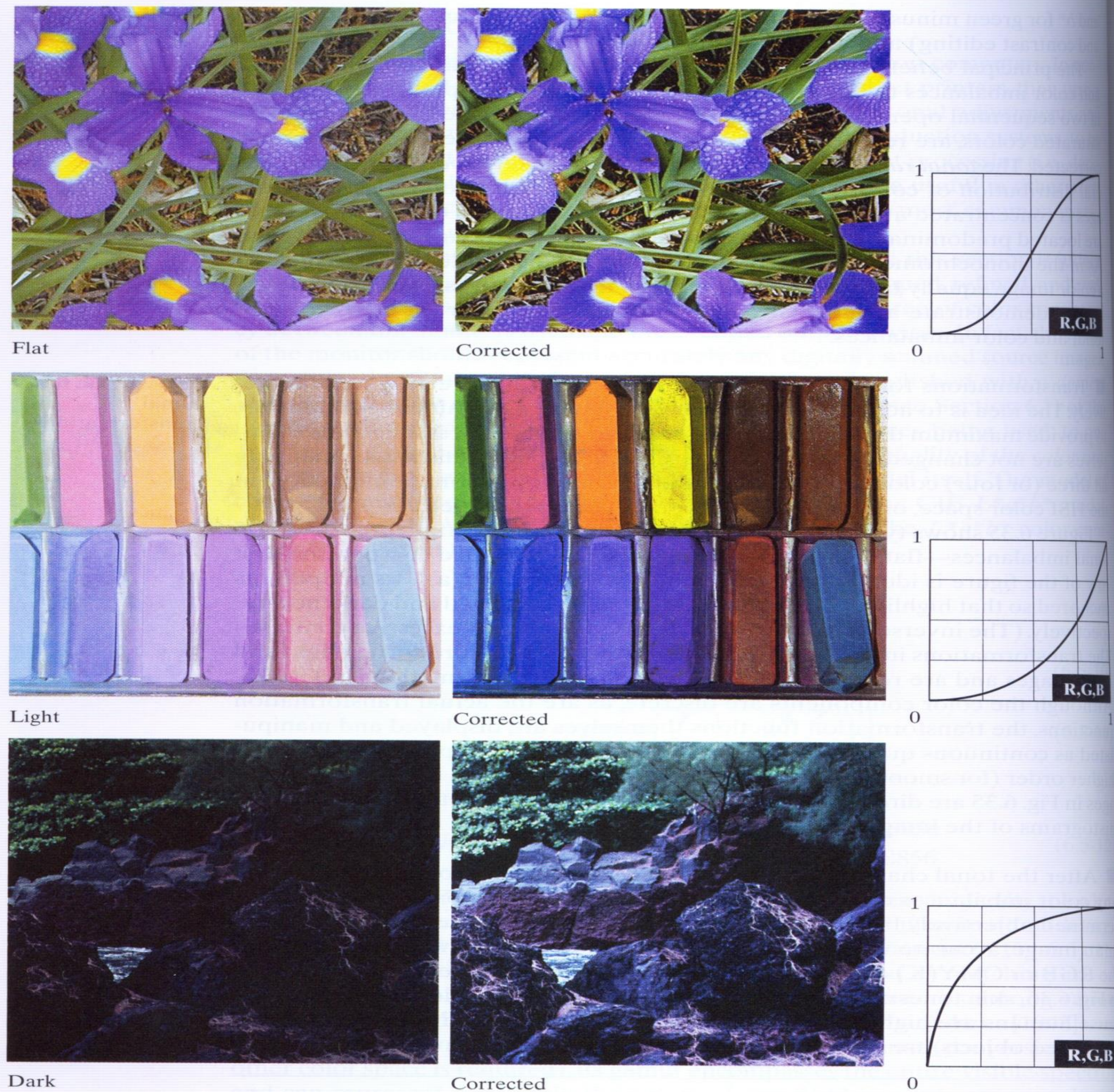


FIGURE 6.35 Tonal corrections for flat, light (high key), and dark (low key) color images. Adjusting the red, green, and blue components equally does not alter the image hues.

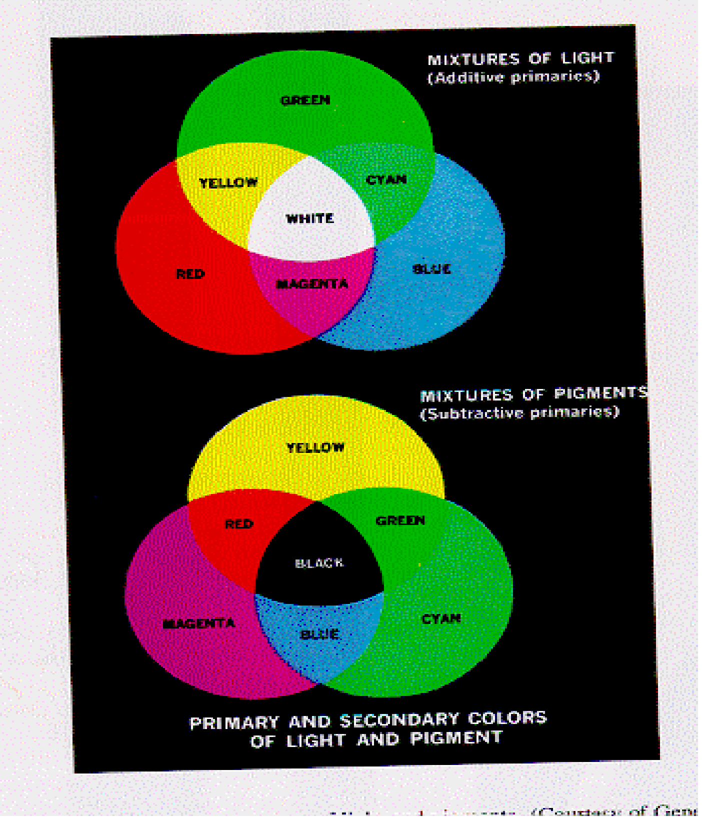
CMYK Color System

- Cyan (255,0,0), Magenta (0,255,0), Yellow (0,0,255), Red (0,255,255), Green (255,0,255), Blue (255,255,0), etc.
- A practical variant, CMYK (with K standing for "black"), is spawned to provide an inexpensive ink. It is called 4-color printing.

K

$$\begin{array}{c} \textit{cyan} \\ \textit{magenta} \\ \textit{yellow} \end{array} \begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

(assume all color values have been normalized to the range [0,1])



Color Balancing Corrections for CMYK Color Images

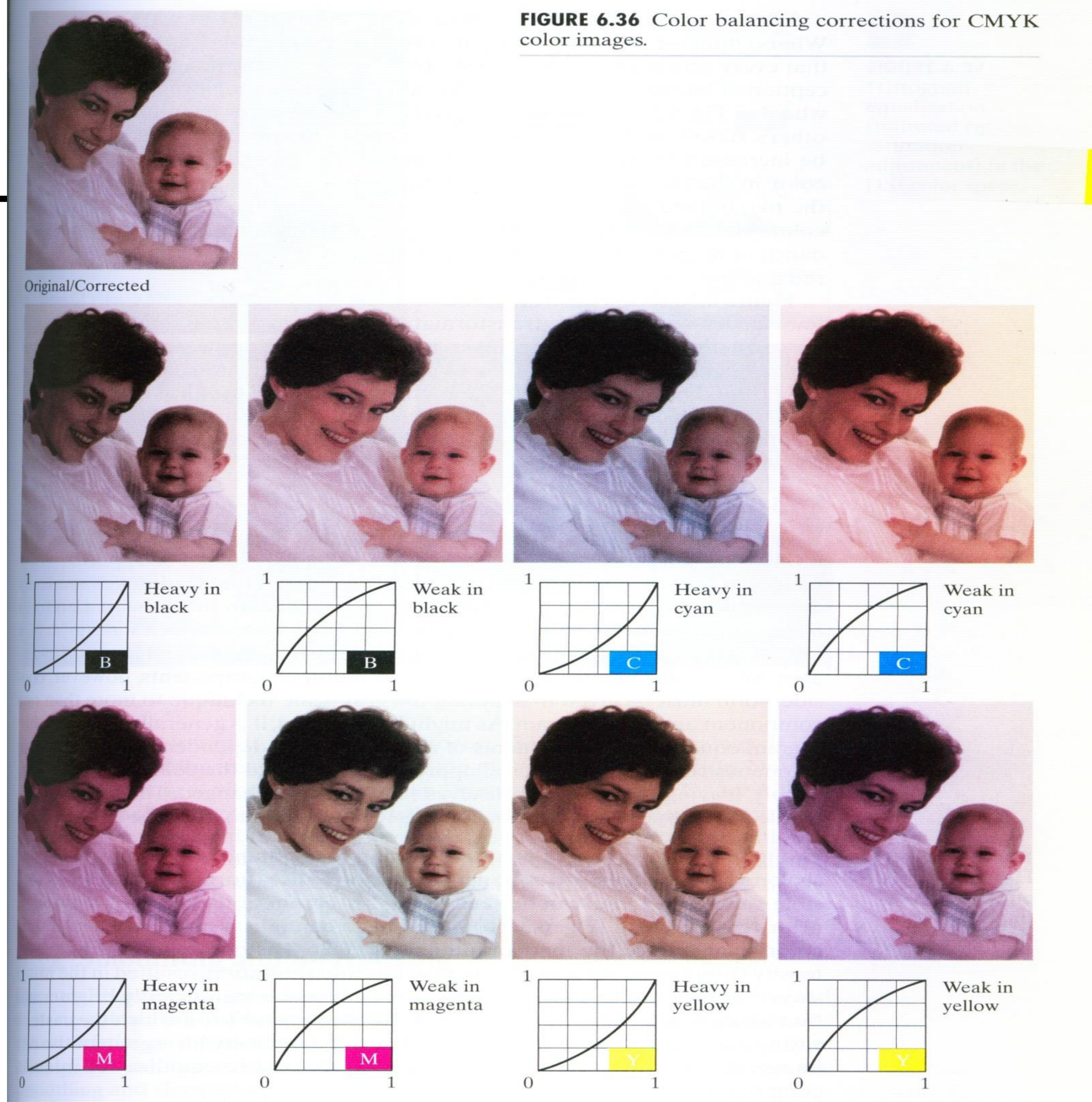
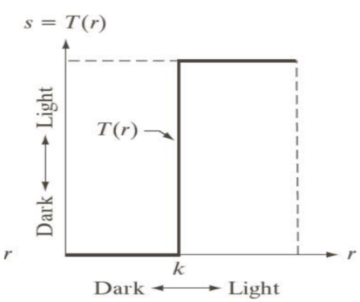
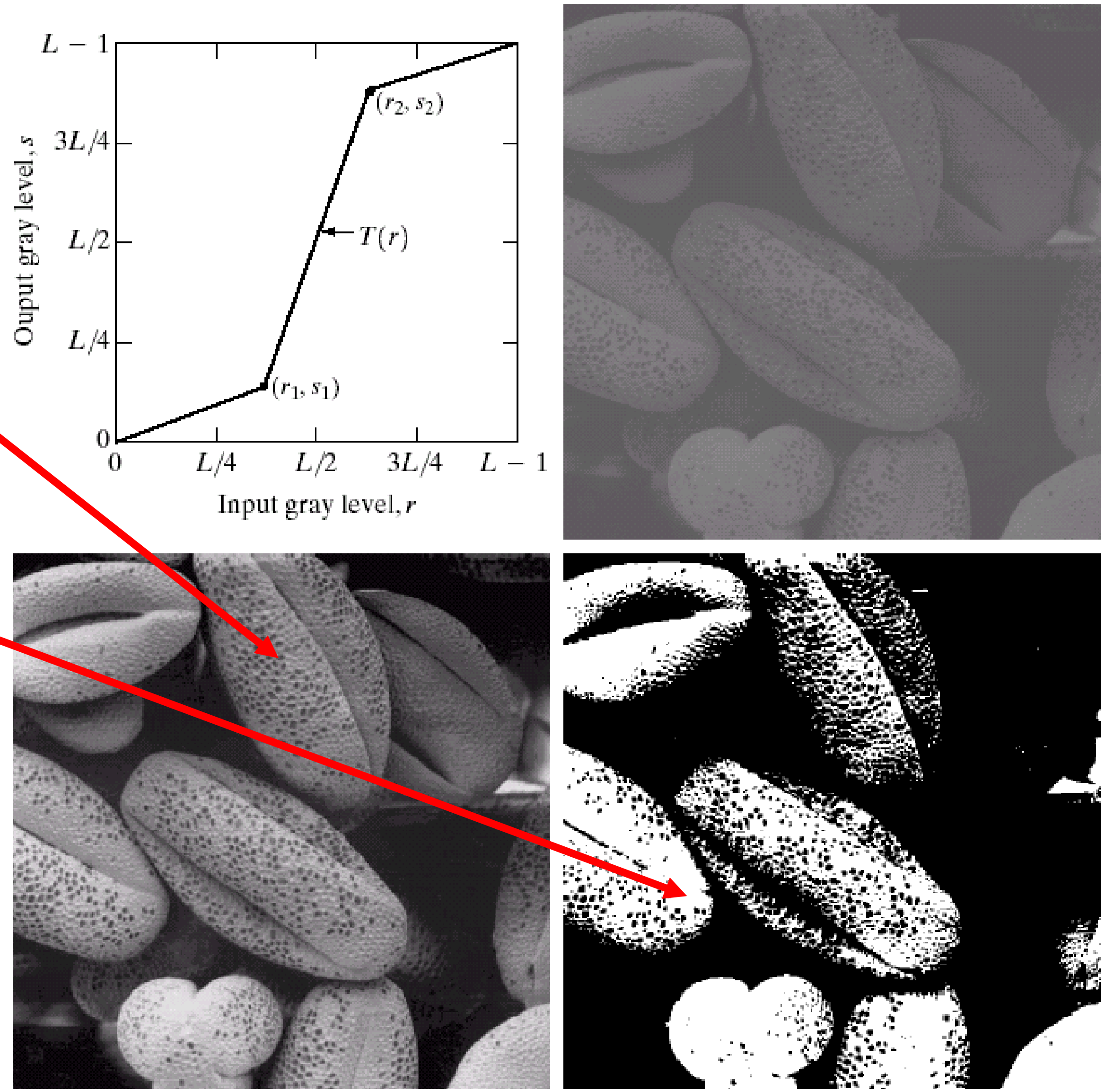


FIGURE 6.36 Color balancing corrections for CMYK color images.

Gray Level Transformations: Contrast Stretching

It is normally controlled by a piecewise "linear transformation". A special case is the "thresholding" when r1=r2 and s1=0, s2=L-1.



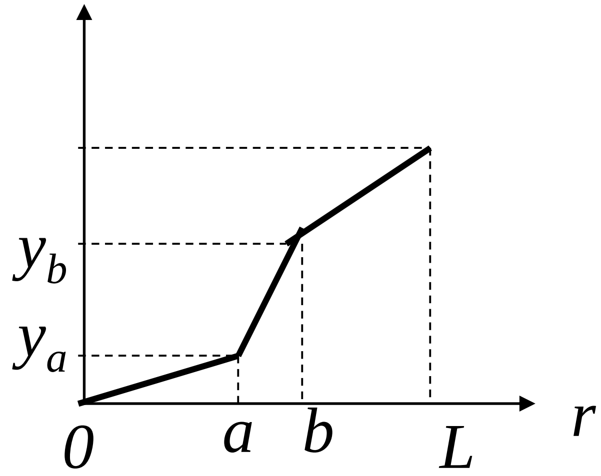


a b
c d

FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

Contrast Stretching

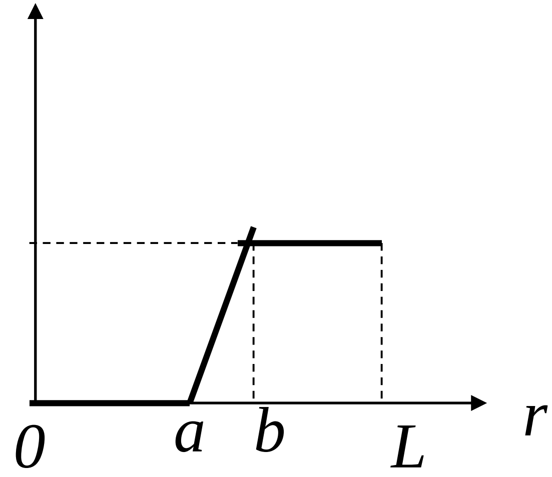
$$T(r) = \begin{cases} \alpha r & 0 \le r < a \\ \beta(r-a) + y_a & a \le r < b \\ \gamma(r-b) + y_b & b \le r < L \end{cases}$$



$a = 50, b = 150, \alpha = 0.2, \beta = 2, \gamma = 1, y_a = 30, y_b = 200$

Image Intensity Clipping

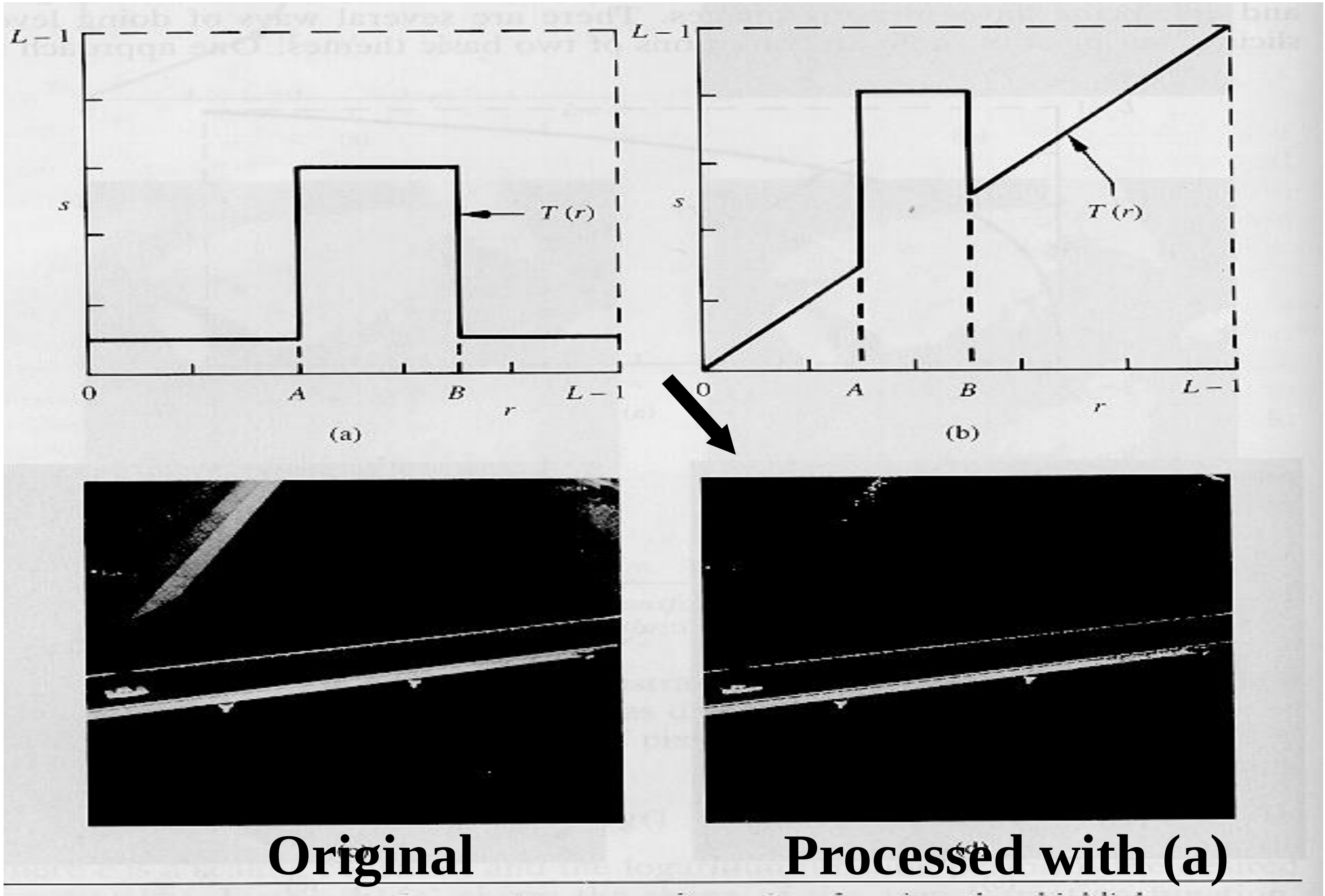
$$T(r) = \begin{cases} 0 & 0 \le r < a \\ \beta(r-a) & a \le r < b \\ \beta(r-a) & b \le r < L \end{cases}$$



$a = 50, b = 150, \beta = 2$

Point Processing: Gray-Level Slicing

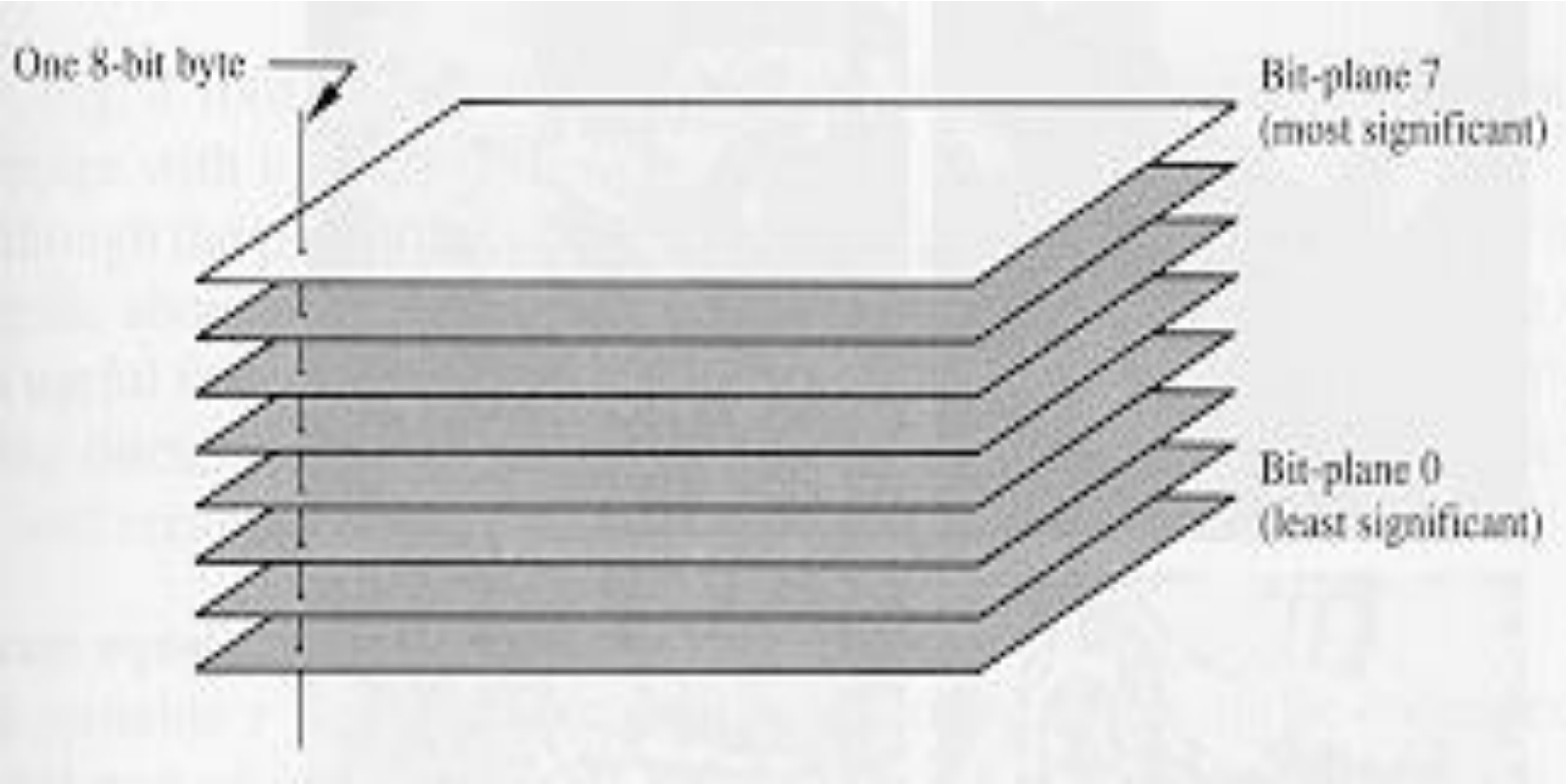
- **a) display a “high value” for gray levels of interest and a low value for others,**
  b) brighten the desired range of gray levels but preserve the background.



Original **Processed with (a)**

Point Processing: Bit-Plane Slicing

- It highlights the contributions made to the total image appearance by specific bits.



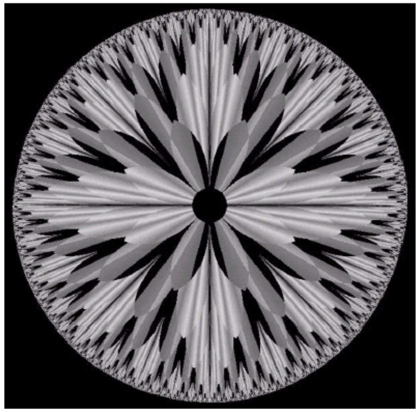


FIGURE 3.13 An 8-bit fractal image. (A fractal is an image generated from m
expressions). (Courtesy of Ms. Melissa D. Binde, Swarthmore College, Swart

Point Processing: Bit-Plane Slicing

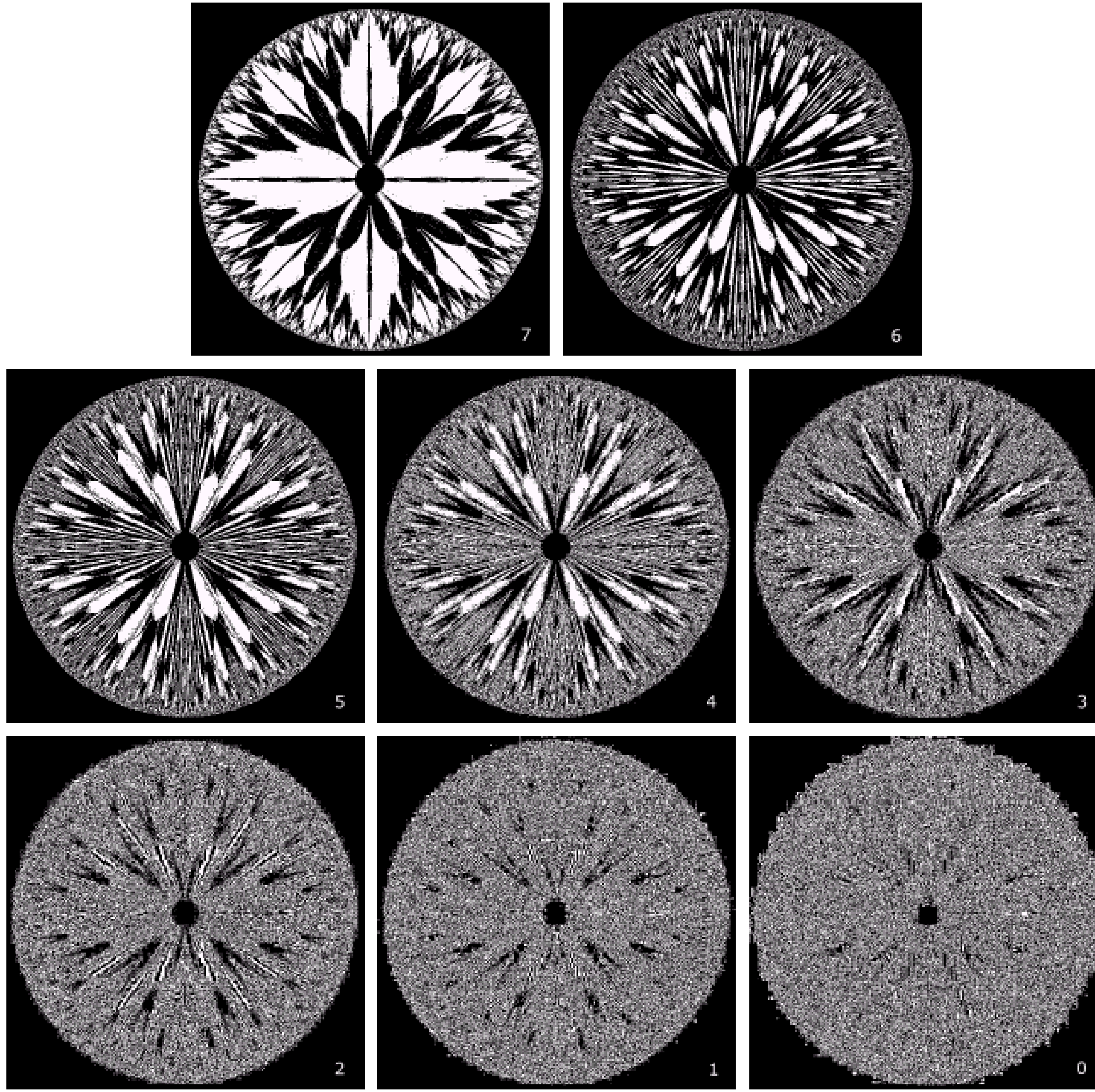


FIGURE 3.14 The eight bit planes of the image in Fig. 3.13. The number at the bottom, right of each image identifies the bit plane.

Point Processing: Bit-Plane Slicing

FIGURE 3.14 (a) An 8-bit gray-scale image of size 500 × 1192 pixels. (b) through (i) Bit planes 1 through 8, with bit plane 1 corresponding to the least significant bit. Each bit plane is a binary image.

Combining Bitplanes

a b c

FIGURE 3.15 Images reconstructed using (a) bit planes 8 and 7; (b) bit planes 8, 7, and 6; and (c) bit planes 8, 7, 6, and 5. Compare (c) with Fig. 3.14(a).

Pseudo-Color Image Processing: Intensity Slicing

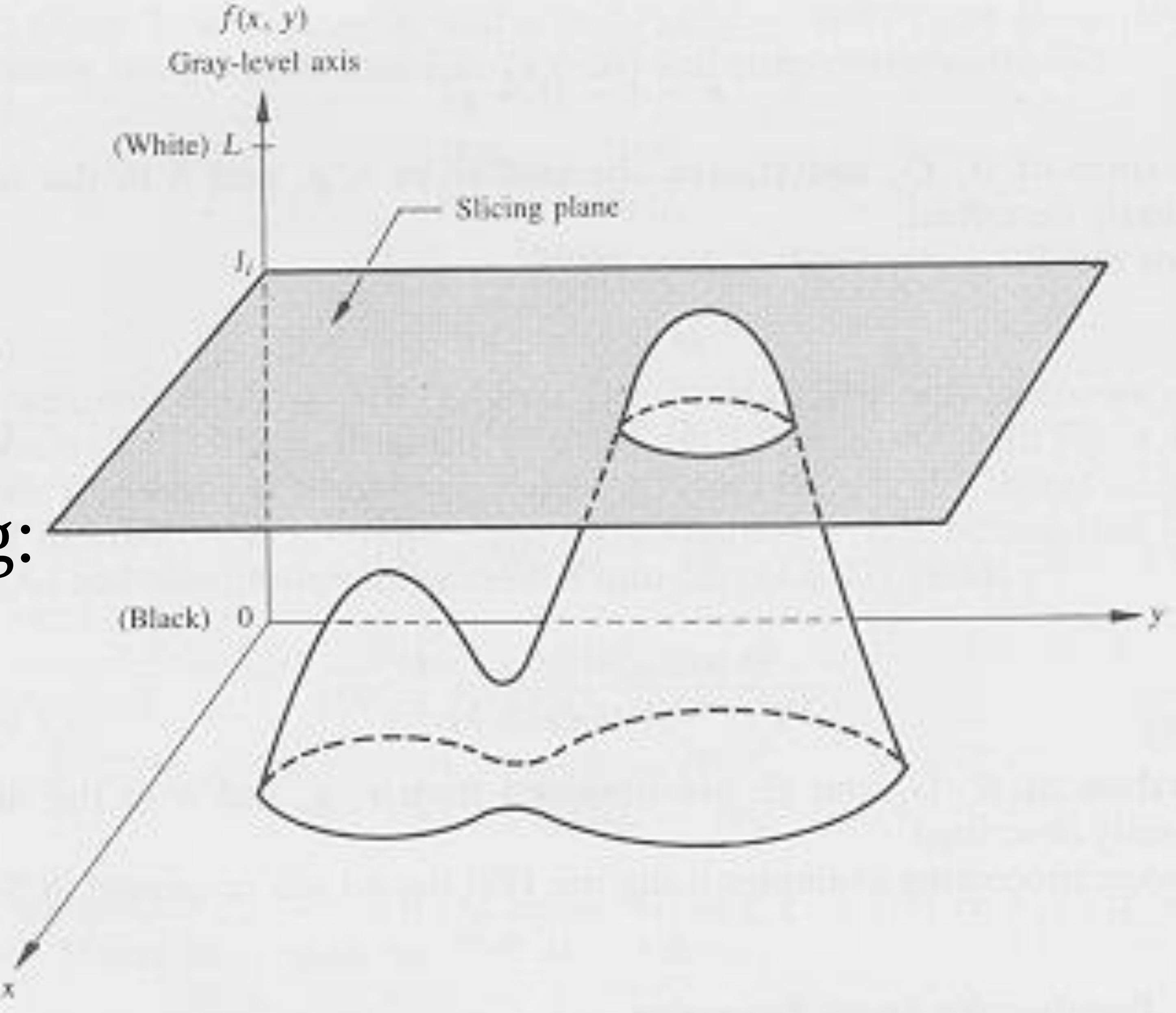
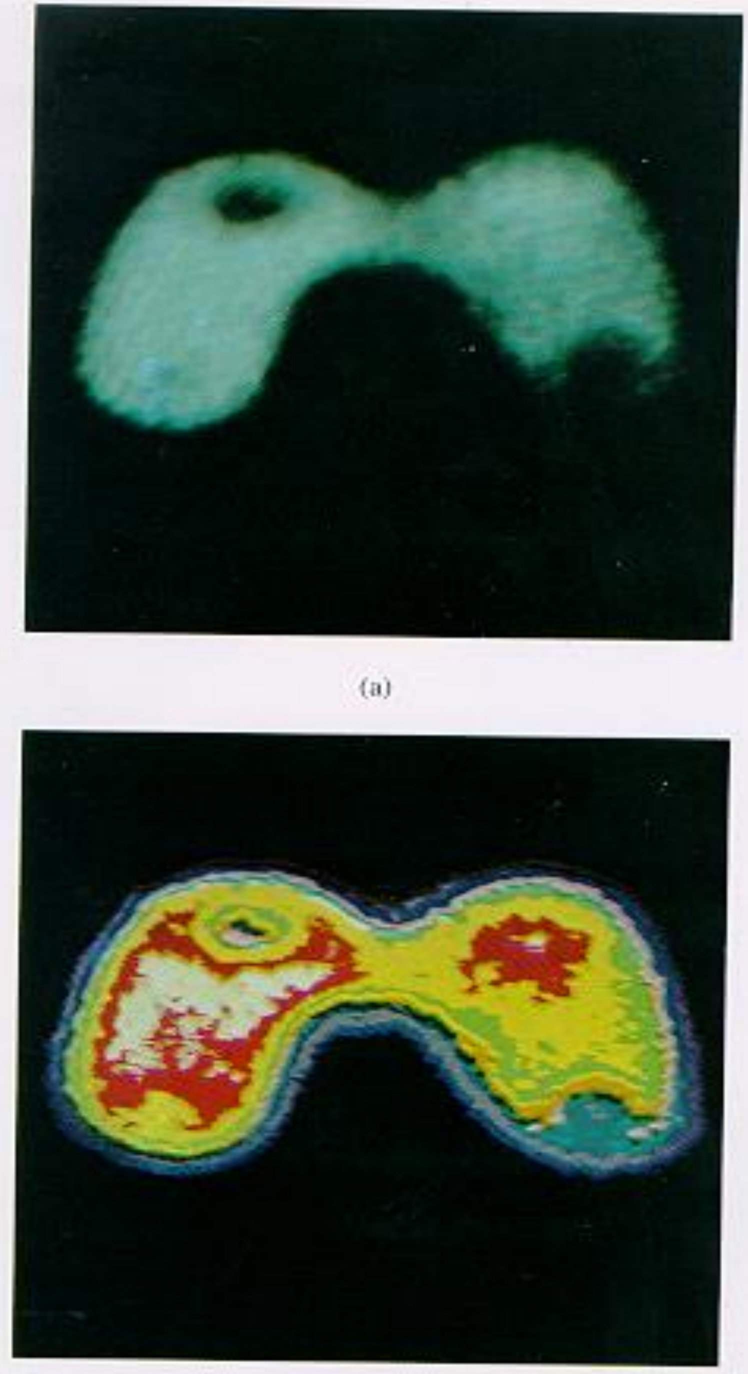


Figure 4.48 Geometric interpretation of the intensity-slicing technique.

Pseudo-Color Image Processing: Intensity Slicing



(a)

(b)

What is a Histogram?

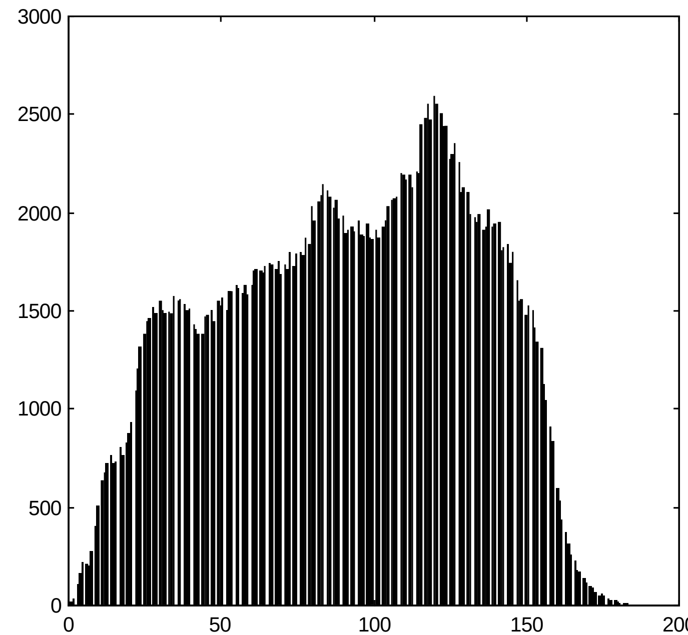
- In Statistics, **Histogram** is a graphical representation showing a visual impression of the distribution of data.
- An **Image Histogram** is a type of histogram that acts as a graphical representation of the lightness/color distribution in a digital image. It plots the number of pixels for each value (bin).
- The histogram of a digital image with gray levels in the range $[0, L-1]$ is a discrete function $h(r_k) = n_k$, where r_k is the k-th gray level and n_k is the number of pixels in the image having gray level r_k.

Histogram Processing

- The number of bins can be smaller than the number of gray levels, e.g., every 4 levels form one bin (256 gray levels, 64 bins).
- It is common practice to normalize a histogram by dividing each of its values by the total number of pixels in the image, denoted by n. Thus, a normalized histogram is given by $p(r_k) = n_k / n$, for $k = 0, 1, \ldots, L-1$.
- Thus, $p(r_k)$ gives an estimate of the probability of density (occurrence) of gray level r_k. Note that the sum of all components of a normalized histogram is equal to 1.

Histogram & Probability of Density

MATLAB
imhist(x)

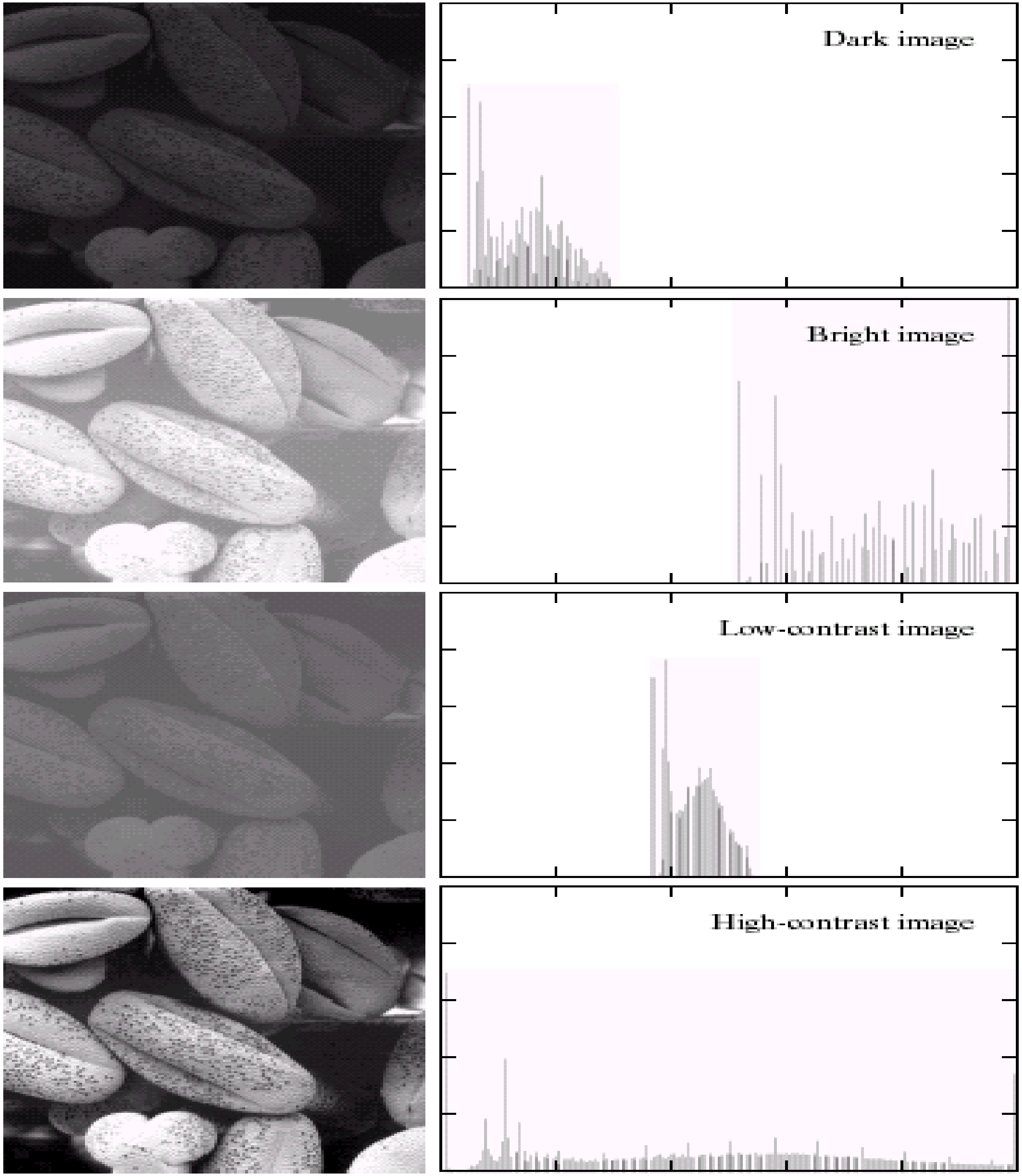


r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

for an M-by-N (64x64) image size

Image Histogram

Histograms of different kinds of images



Histogram Equalization (HE)

- Basic idea: find a map *T(r)* such that the histogram (probability of density, pdf) of the modified (equalized) image is flat (uniform).

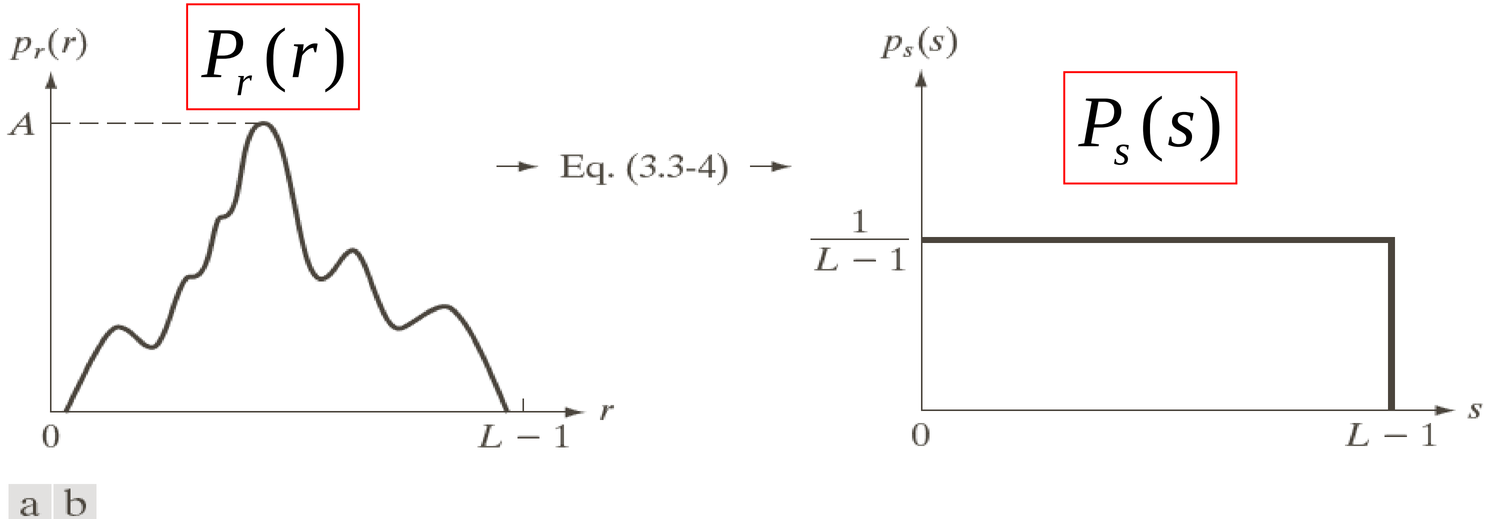


FIGURE 3.18 (a) An arbitrary PDF. (b) Result of applying the transformation in Eq. (3.3-4) to all intensity levels, r. The resulting intensities, s, have a uniform PDF, independently of the form of the PDF of the r's.

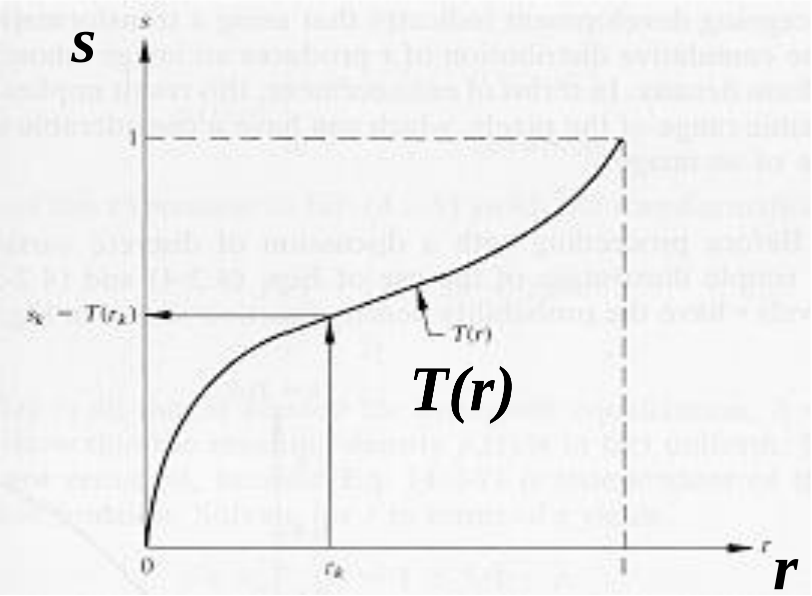
Histogram Equalization (HE)

- Key motivation: cumulative probability function (cdf) of a random variable approximates a uniform distribution

Suppose *Pr(k)* is the histogram (pdf)

$$0 \le T(r) \le 1, \quad 0 \le r \le 1$$
$$0 \le T^{-1}(s) \le 1, \quad 0 \le s \le 1$$

$$\boxed{s = T(r) = \sum_{k=0}^{r} \Pr(k)}$$



Histogram Equalization

Let $\dfrac{ds}{dr} = p_r(r)\ ,\ 0 \le r \le 1$

$p_r(\,)$ is the probability density function.

Then $\boxed{s = T(r) = \int_0^r p_r(\omega)d\omega}$.

$$p_s(s) = \left[\, p_r(r)\frac{dr}{ds} \right]_{r=T^{-1}(s)}$$

$$\boxed{\frac{dr}{ds} = \frac{1}{p_r(r)}}$$

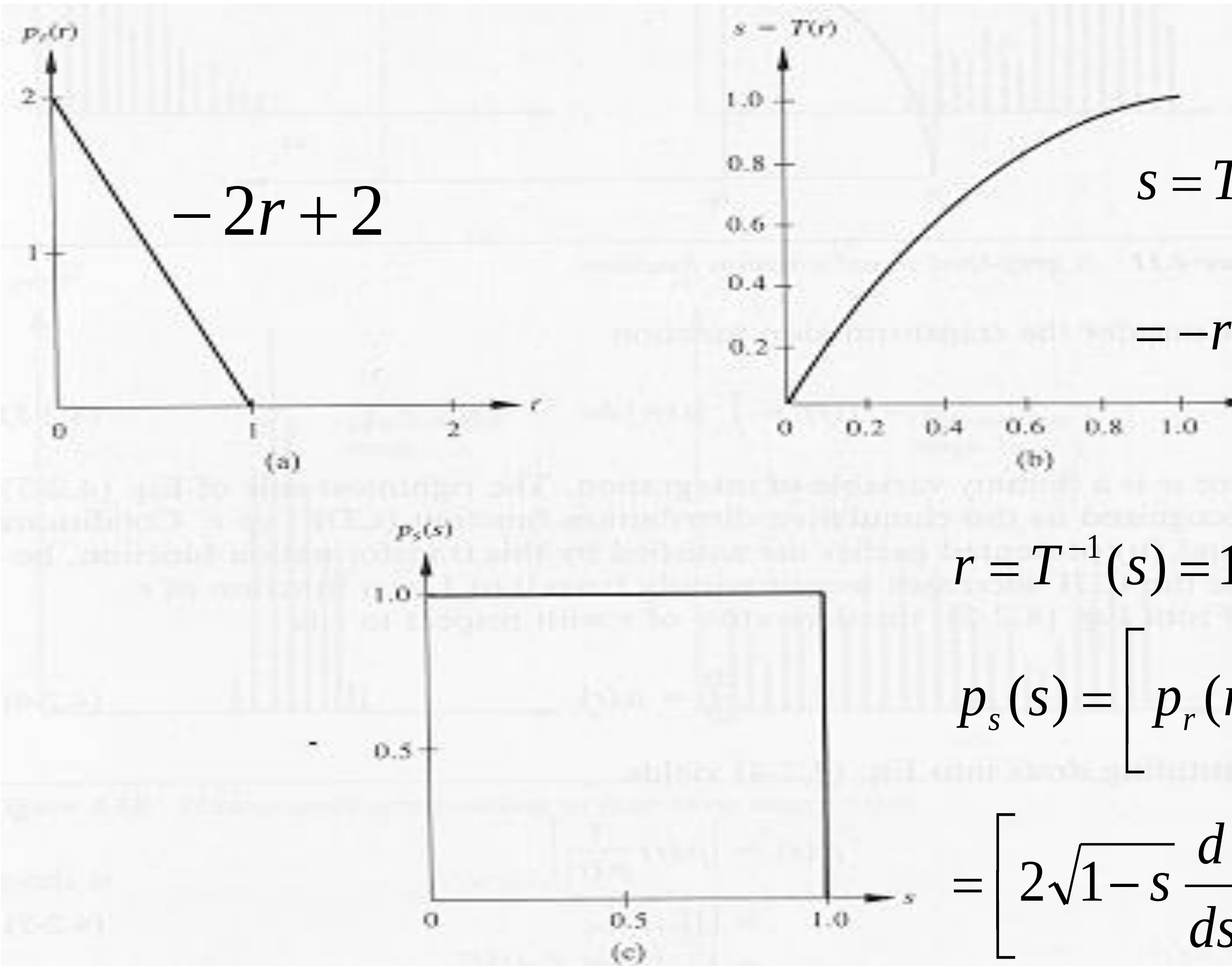
$$\Rightarrow\ p_s(s) = 1, \quad 0 \le s \le 1$$

For digital images, r_k is discrete:

$$p_r(r_k) = \frac{n_k}{N}$$

$$s_k = T(r_k) = \boxed{\sum_{j=0}^{k} p_r(r_j) = \sum_{j=0}^{k} \frac{n_j}{N}}$$

Histogram Equalization - Example



$$-2r+2$$

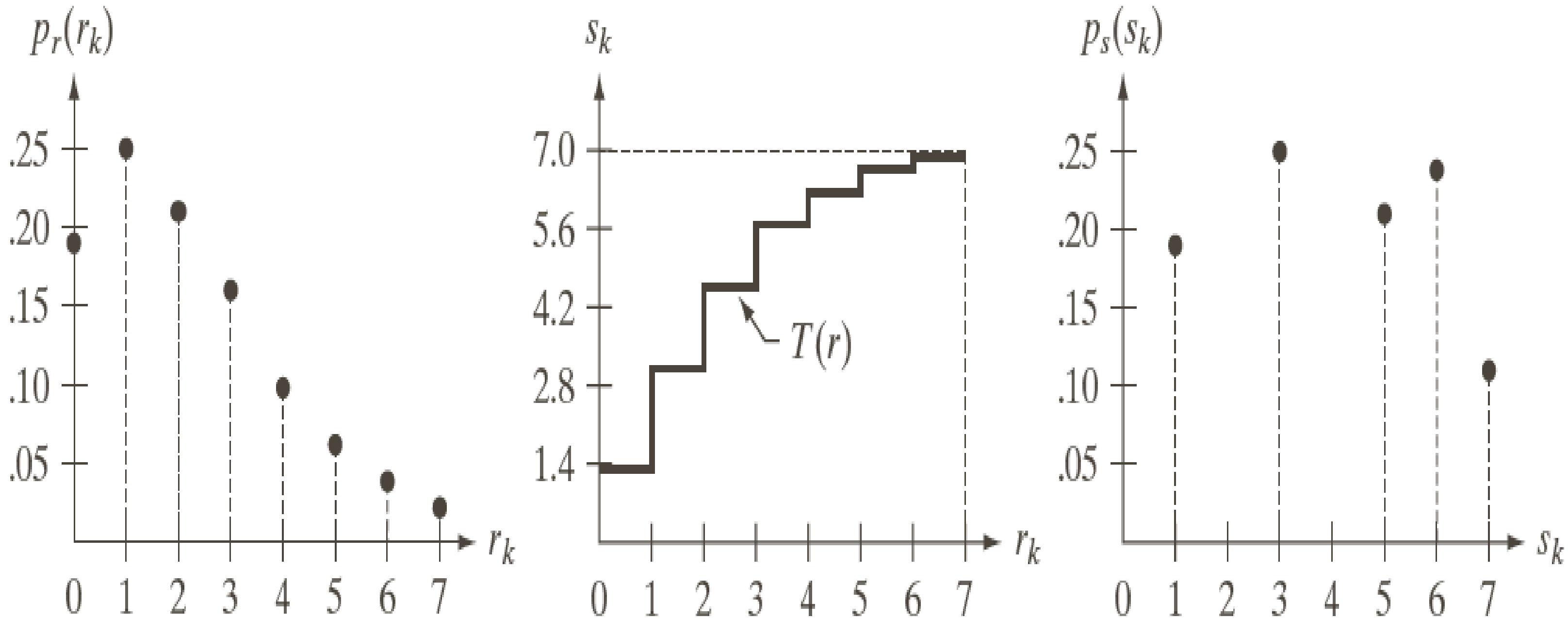
$$s = T(r) = \int_0^r (-2w+2)\,dw$$

$$= -r^2 + 2r$$

$$r = T^{-1}(s) = 1 - \sqrt{1-s}$$

$$p_s(s) = \left[p_r(r) \frac{dr}{ds} \right]_{r=T^{-1}(s)}$$

$$= \left[2\sqrt{1-s}\, \frac{d}{ds}\left(1-\sqrt{1-s}\right) \right] = 1$$

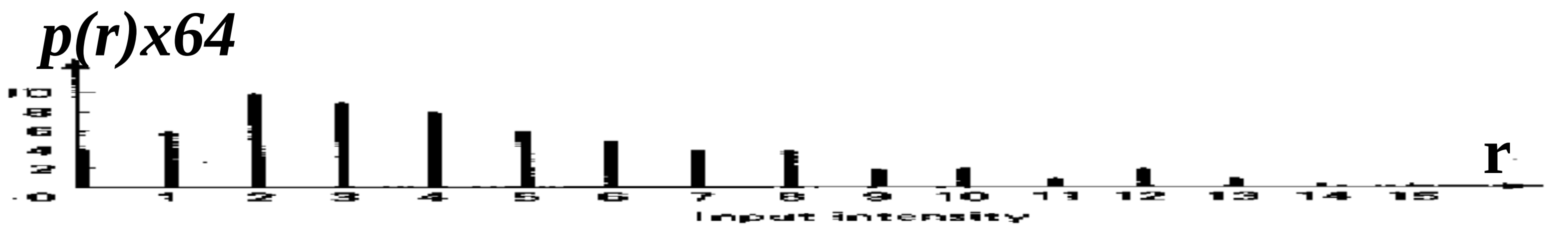


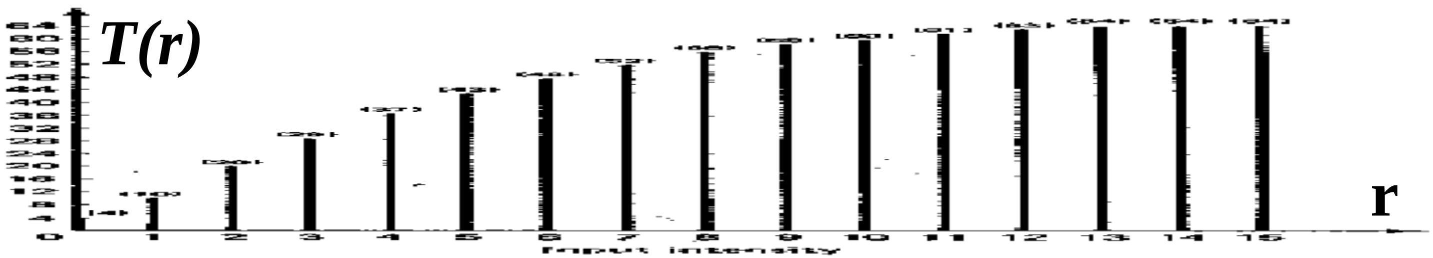
a b c

FIGURE 3.19 Illustration of histogram equalization of a 3-bit (8 intensity levels) image. (a) Original histogram. (b) Transformation function. (c) Equalized histogram.

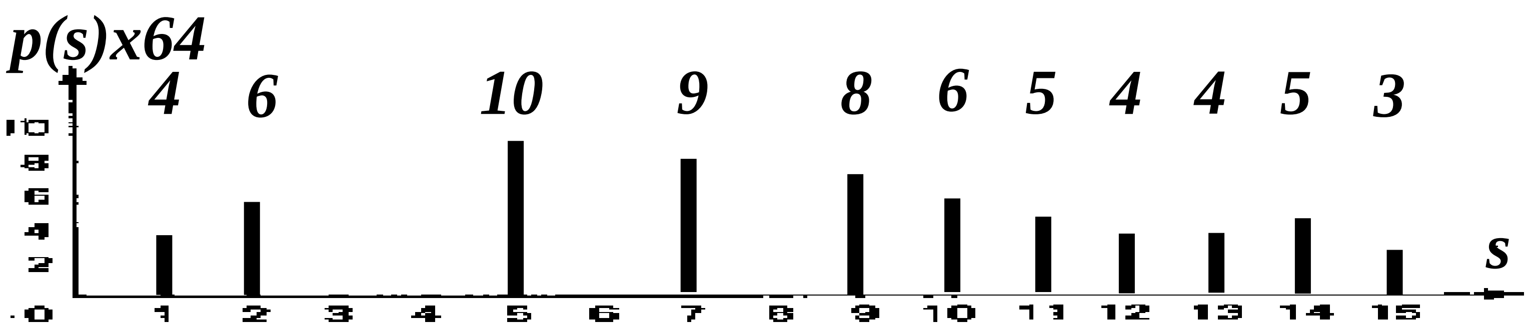
An Example of Histogram Equalization

with $N=64$, $0 <= r <= 15$





r	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*
p(r)x64	*4*	*6*	*10*	*9*	*8*	*6*	*5*	*4*	*4*	*2*	*2*	*1*	*2*	*1*	*0*	*0*
T(r)x64	*4*	*10*	*20*	*29*	*37*	*43*	*48*	*52*	*56*	*58*	*60*	*61*	*63*	*64*	*64*	*64*
s=T(r)x15	*1*	*2*	*5*	*7*	*9*	*10*	*11*	*12*	*13*	*14*	*14*	*14*	*15*	*15*	*15*	*15*



One More Example

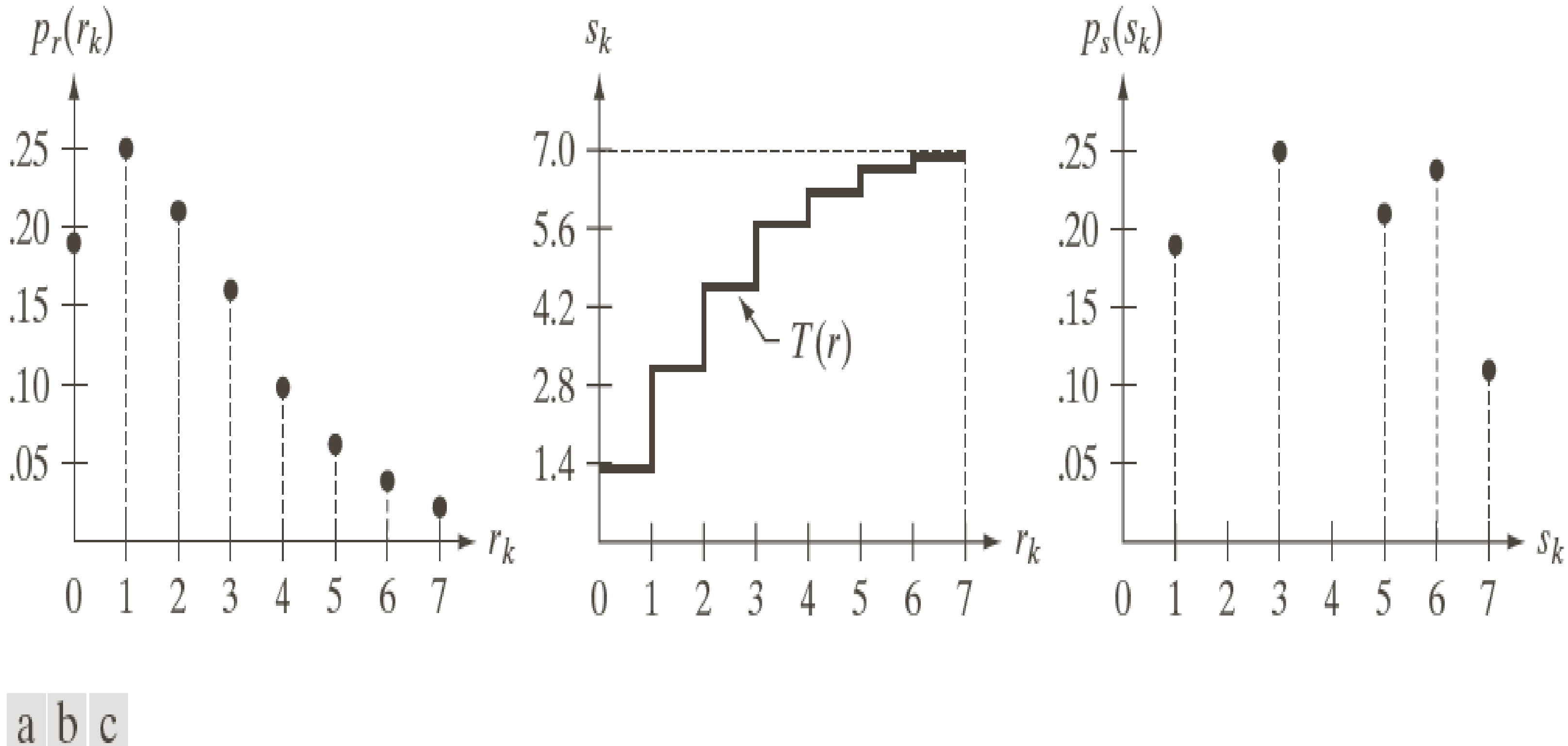
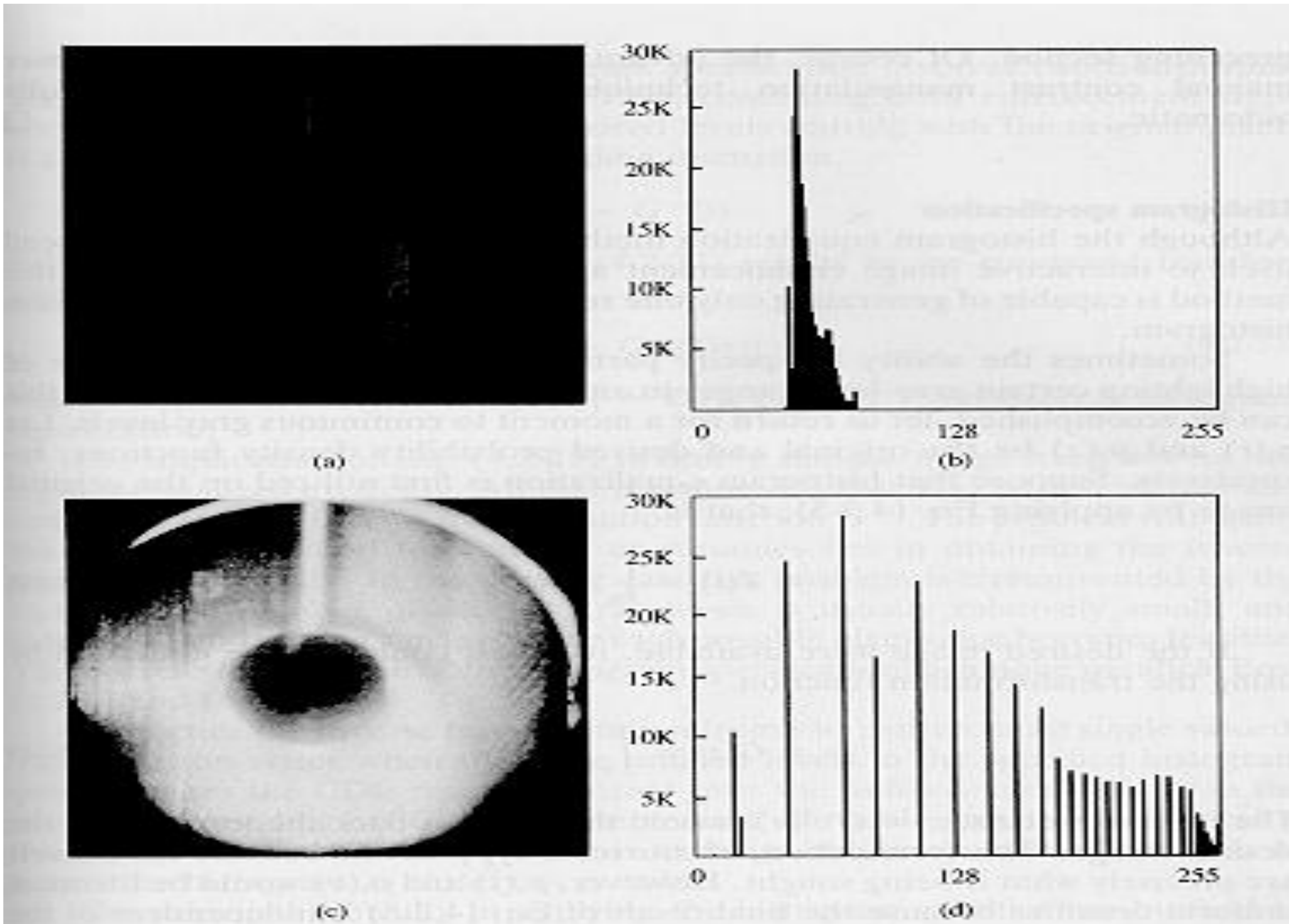
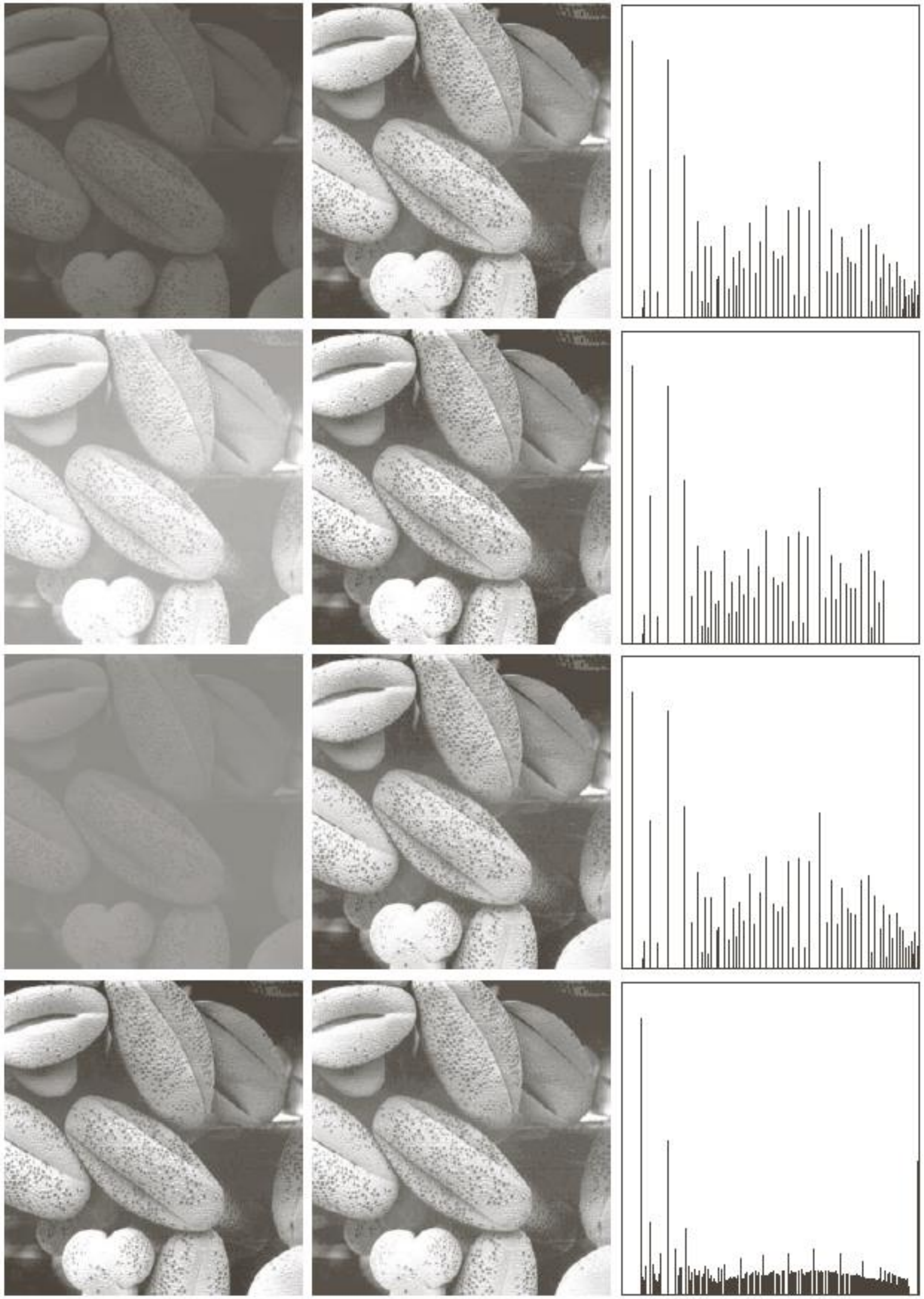


FIGURE 3.19 Illustration of histogram equalization of a 3-bit (8 intensity levels) image. (a) Original histogram. (b) Transformation function. (c) Equalized histogram.

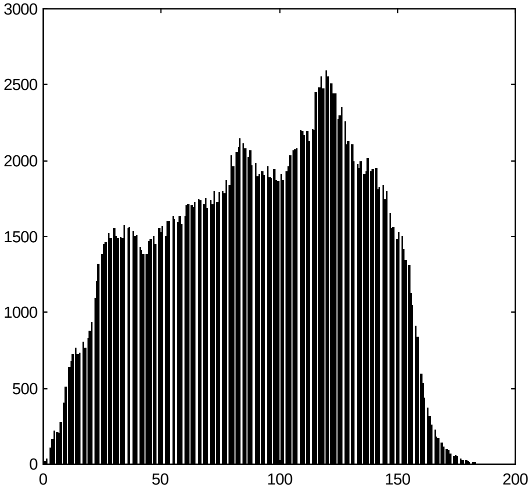
Histogram Equalization (HE) for Contrast Stretching

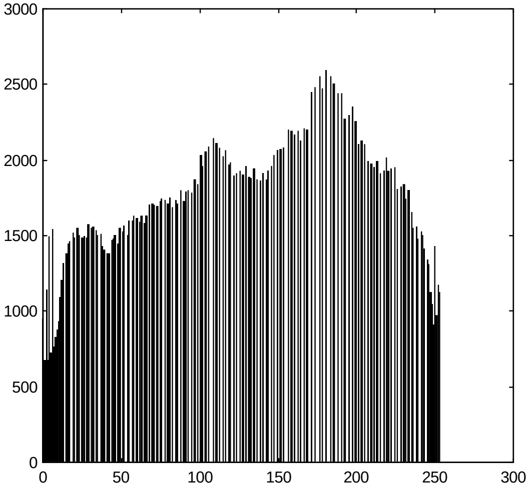




Another HE Example

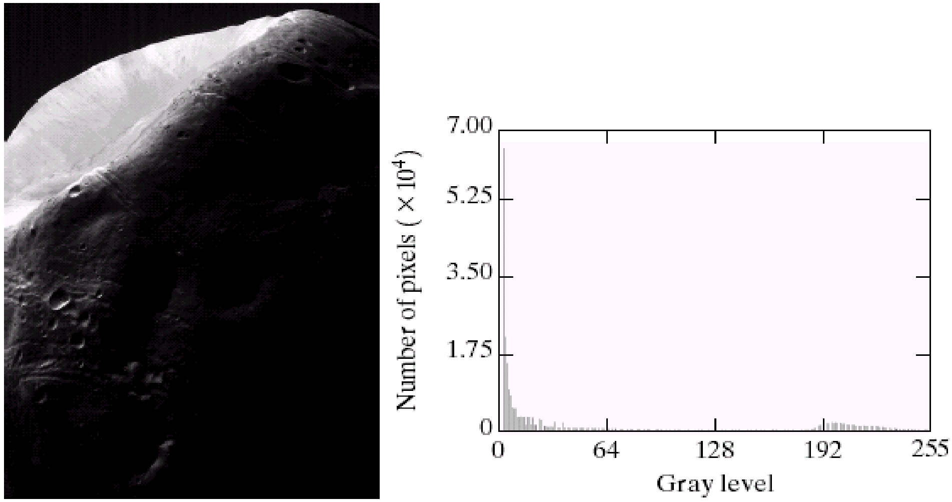
before





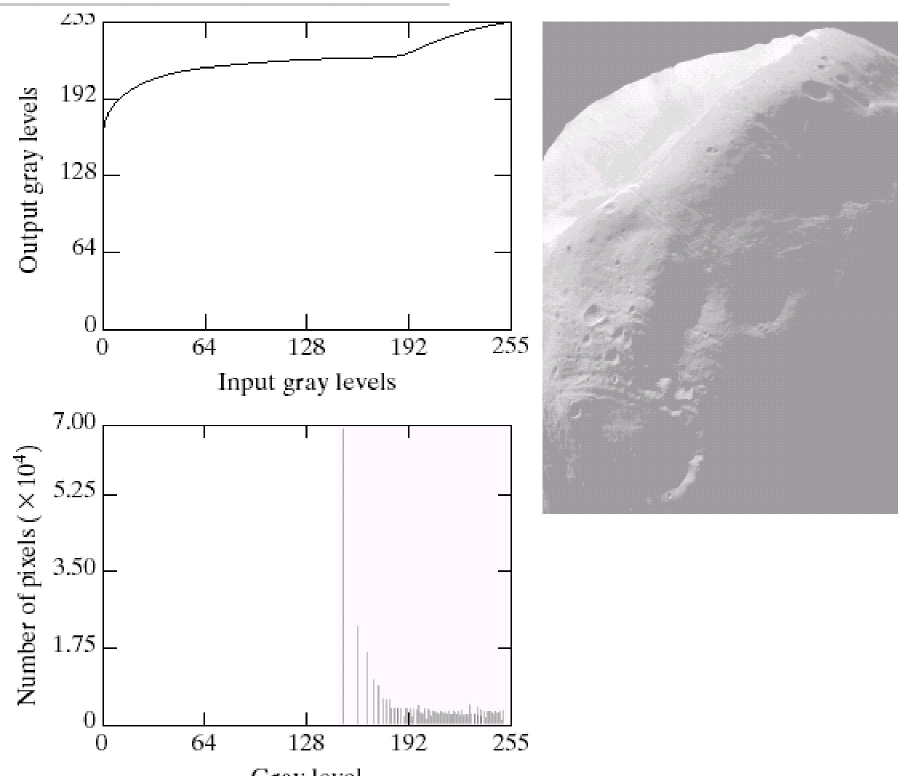
after

Example of Histogram Equalization



a b

FIGURE 3.20 (a) Image of the Mars moon Photos taken by NASA's *Mars Global Surveyor*. (b) Histogram. (Original image courtesy of NASA.)



a b
c

FIGURE 3.21 (a) Transformation function for histogram equalization. (b) Histogram-equalized image (note the washed-out appearance). (c) Histogram of (b).

Histogram Specification

- Histogram equalization tries to generate a uniform histogram.
- For interactive image enhancement, the user may like to result in a customized histogram.

---> Use Histogram Specification

Histogram Specification

An image with pixel values r is to be transformed into an image with a *specified histogram*. Denote the pixel values of the new image as u. Using histogram equalization, we can equalize the original and the desired image to an image with pixel values s with uniform histogram:

$$s = T(r) = \int_0^r p_r(\omega)d\omega$$

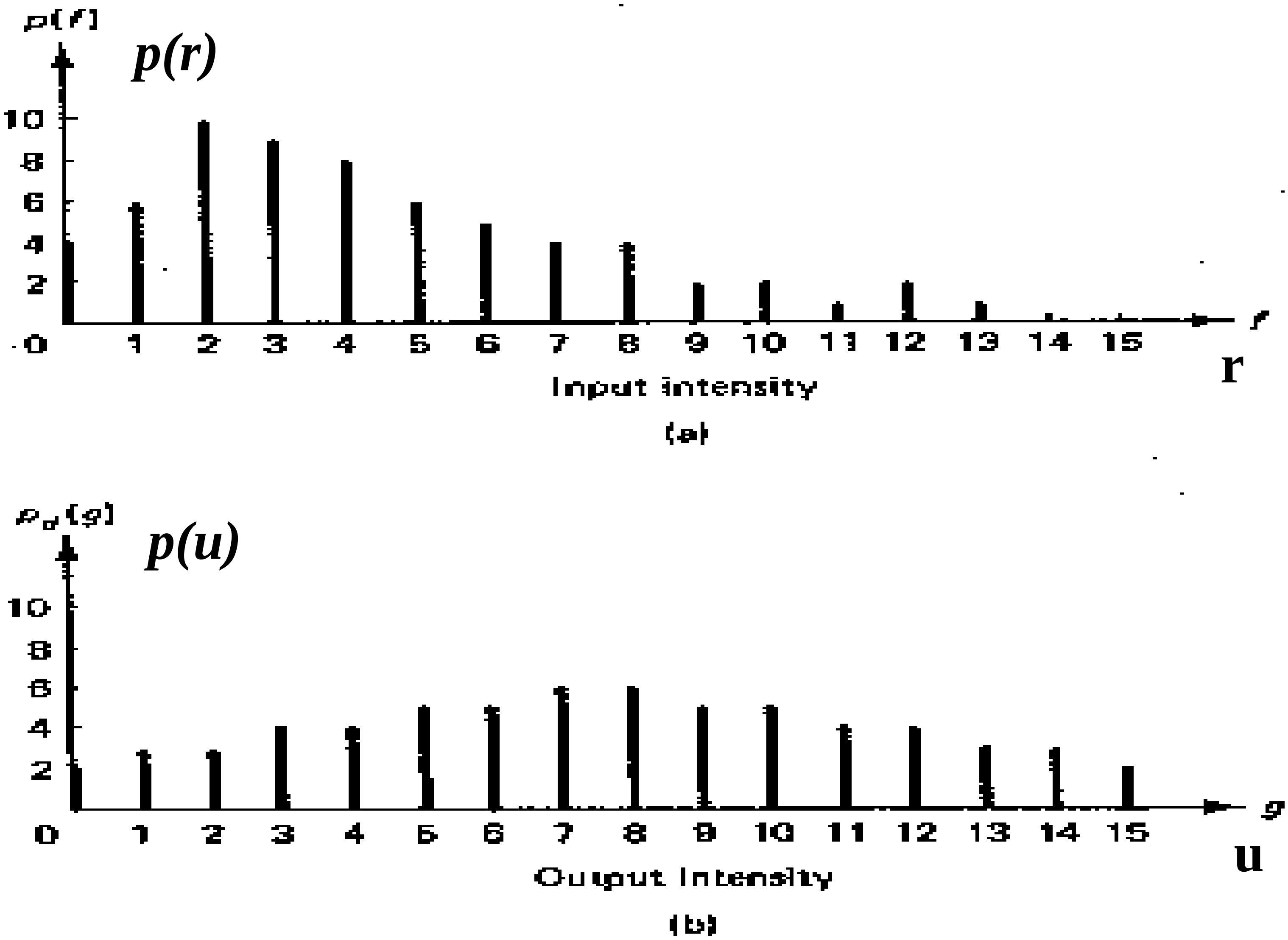
$$s = G(u) = \int_0^u p_u(\omega)d\omega$$

$$u = G^{-1}(s) = G^{-1}(T(r))$$

- **Specify a particular probability density function *p(u)***

=> G(u) => calculate $G^{-1}T$ **for Histogram Specification.**

An Example of Histogram Specification
with N=64, 0 <= r <= 15



An Example of Histogram Specification

with N=64, $0 <= r <= 15$

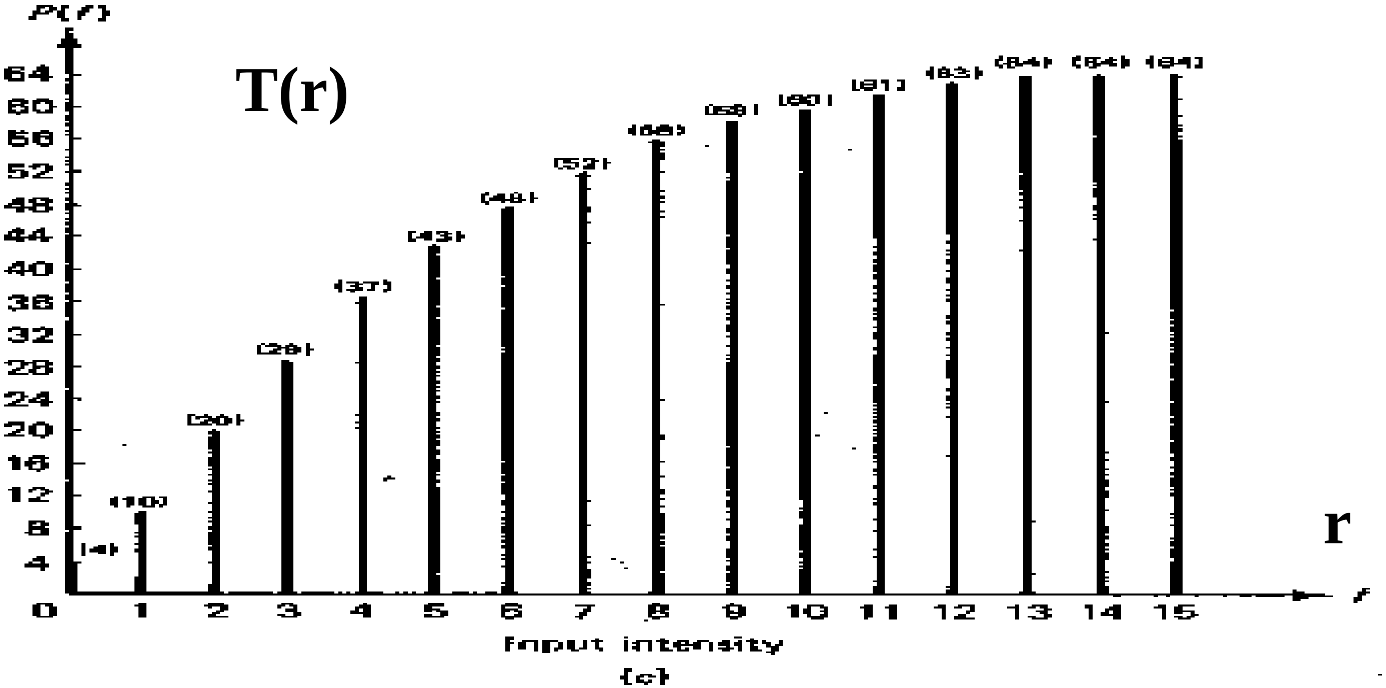
r	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$p(r)$x64	4	6	10	9	8	6	5	4	4	2	2	1	2	1	0	0
$T(r)$x64	4	10	20	29	37	43	48	52	56	58	60	61	63	64	64	64

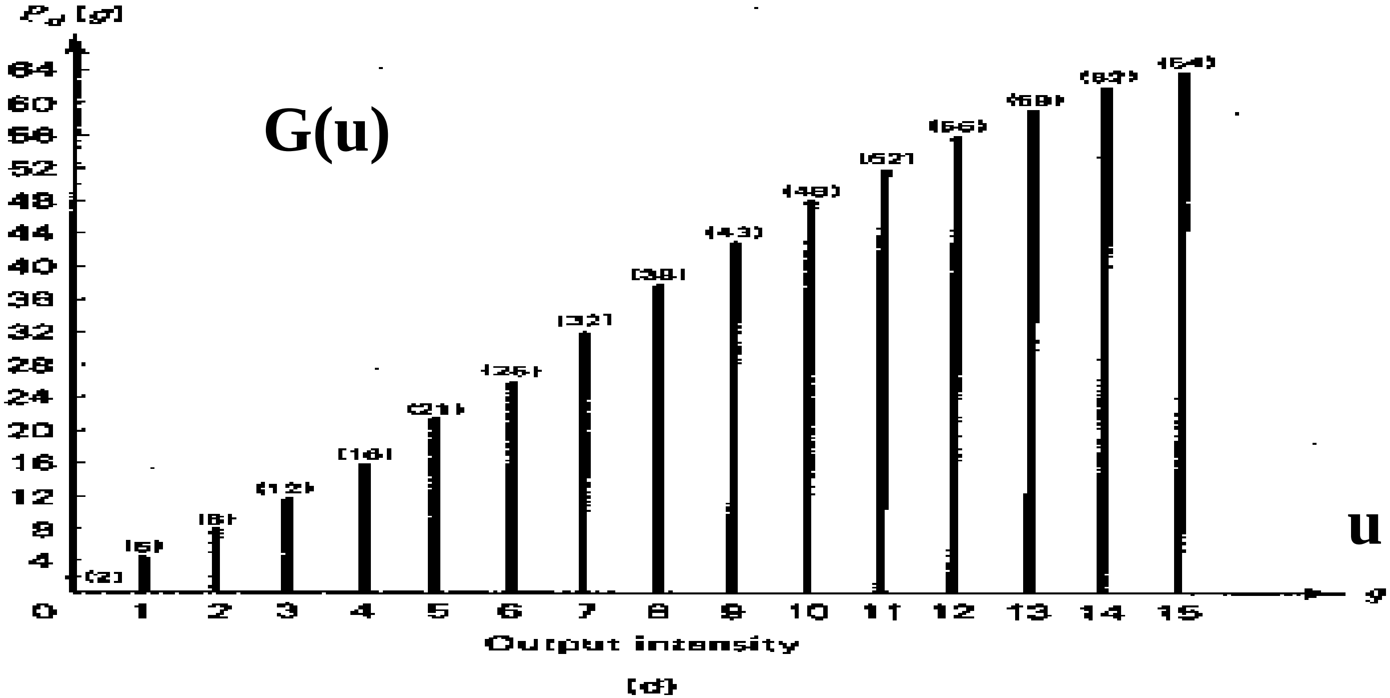
S=T(r

u	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$p(u)$x64	2	3	3	4	4	5	5	6	6	5	5	4	4	3	3	2
$G(u)$x64	2	5	8	12	16	21	26	32	38	43	48	52	56	59	62	64

S=T(u)

r	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$u=G^{-1}(T(r))$x15	1	3	5	7	8	9	10	11	12	13	13	14	14	15	15	15

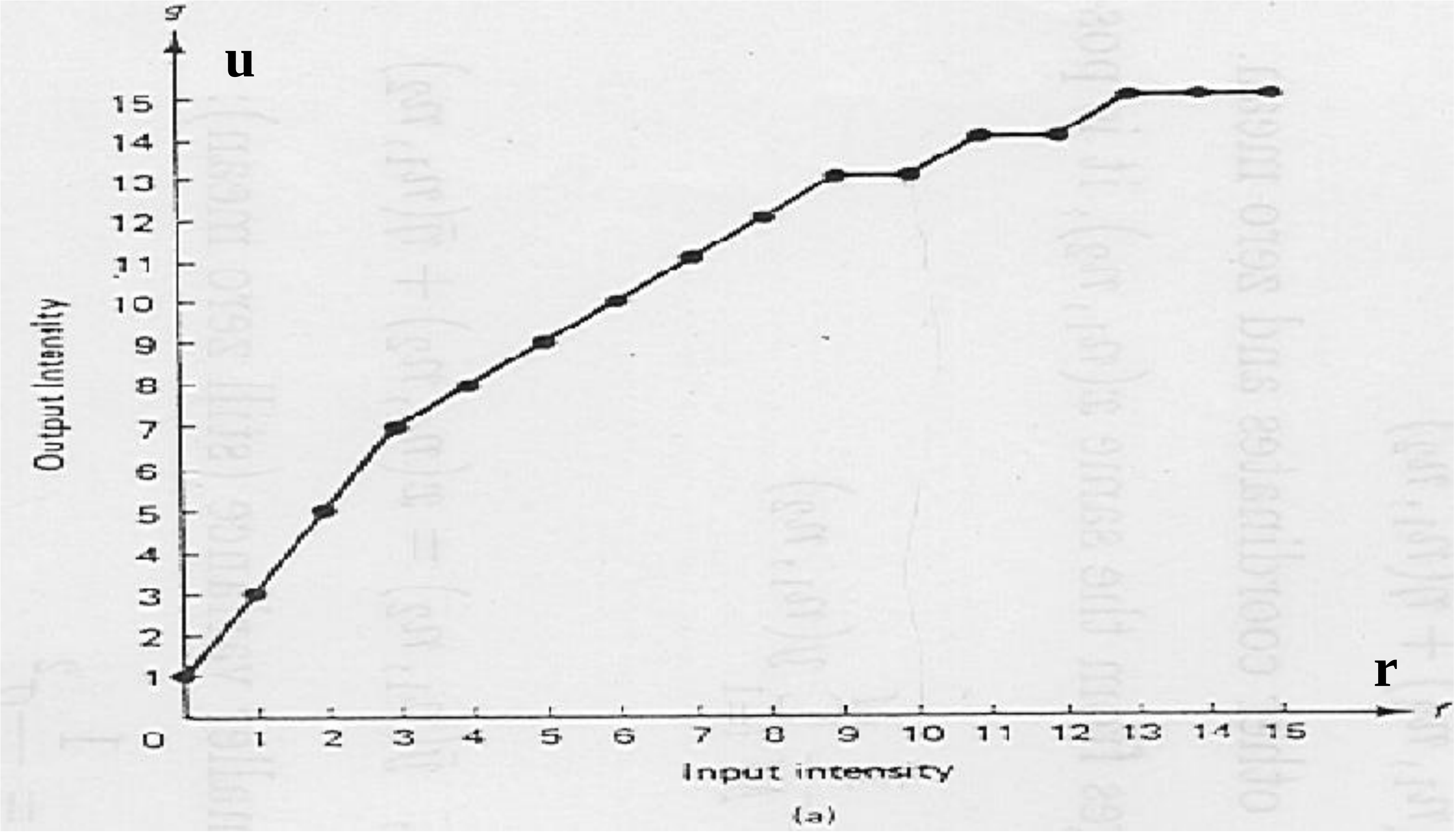

T(r)
r
Input intensity
(c)

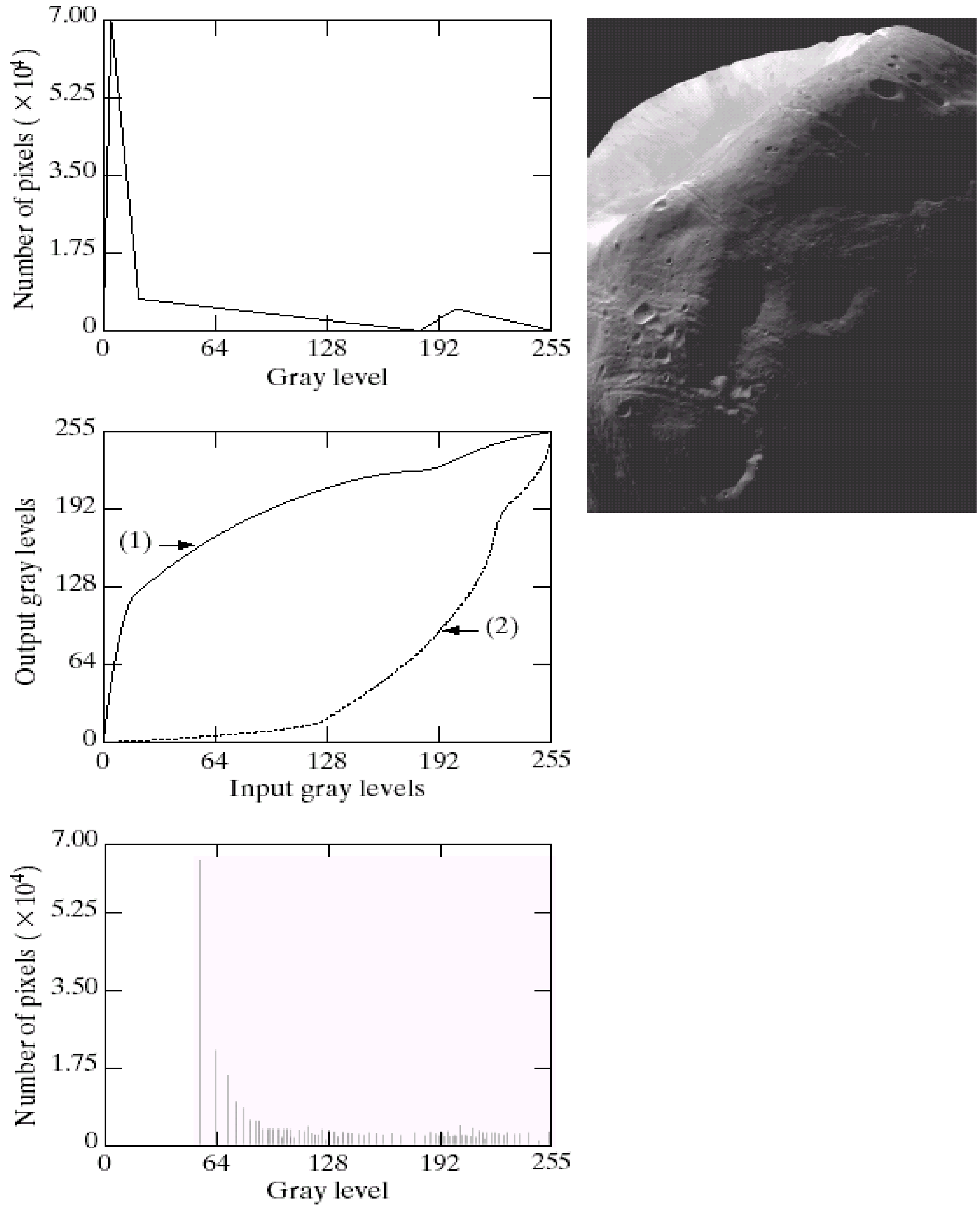

G(u)
u
Output intensity
(d)

An Example of Histogram Specification

with $N=64$, $0 <= r <= 15$ (cot.)

$$u = G^{-1}(T(r))$$



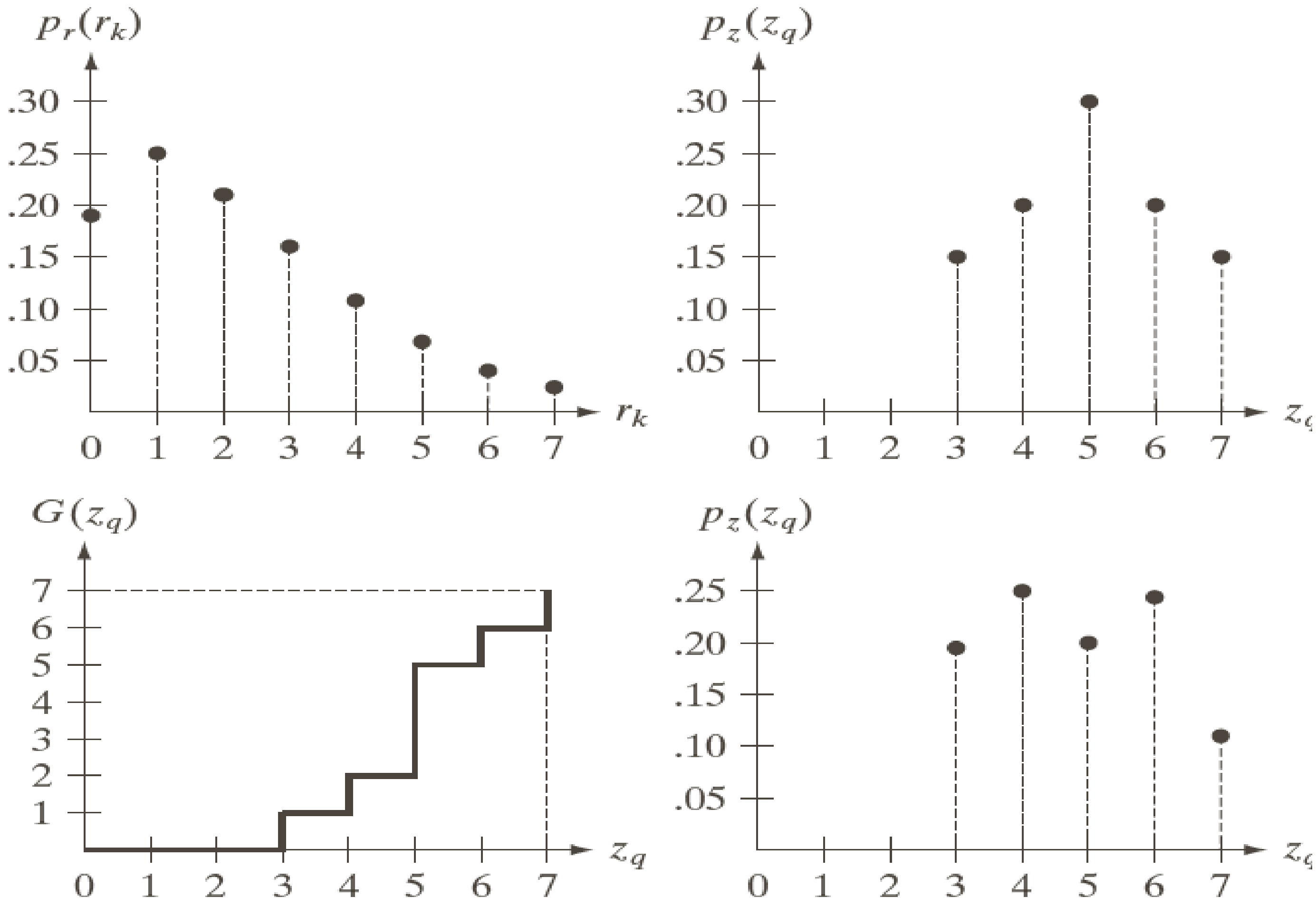


a c
b
d

FIGURE 3.22
(a) Specified histogram.
(b) Curve (1) is from Eq. (3.3-14), using the histogram in (a); curve (2) was obtained using the iterative procedure in Eq. (3.3-17).
(c) Enhanced image using mappings from curve (2).
(d) Histogram of (c).

Example of Histogram Specification

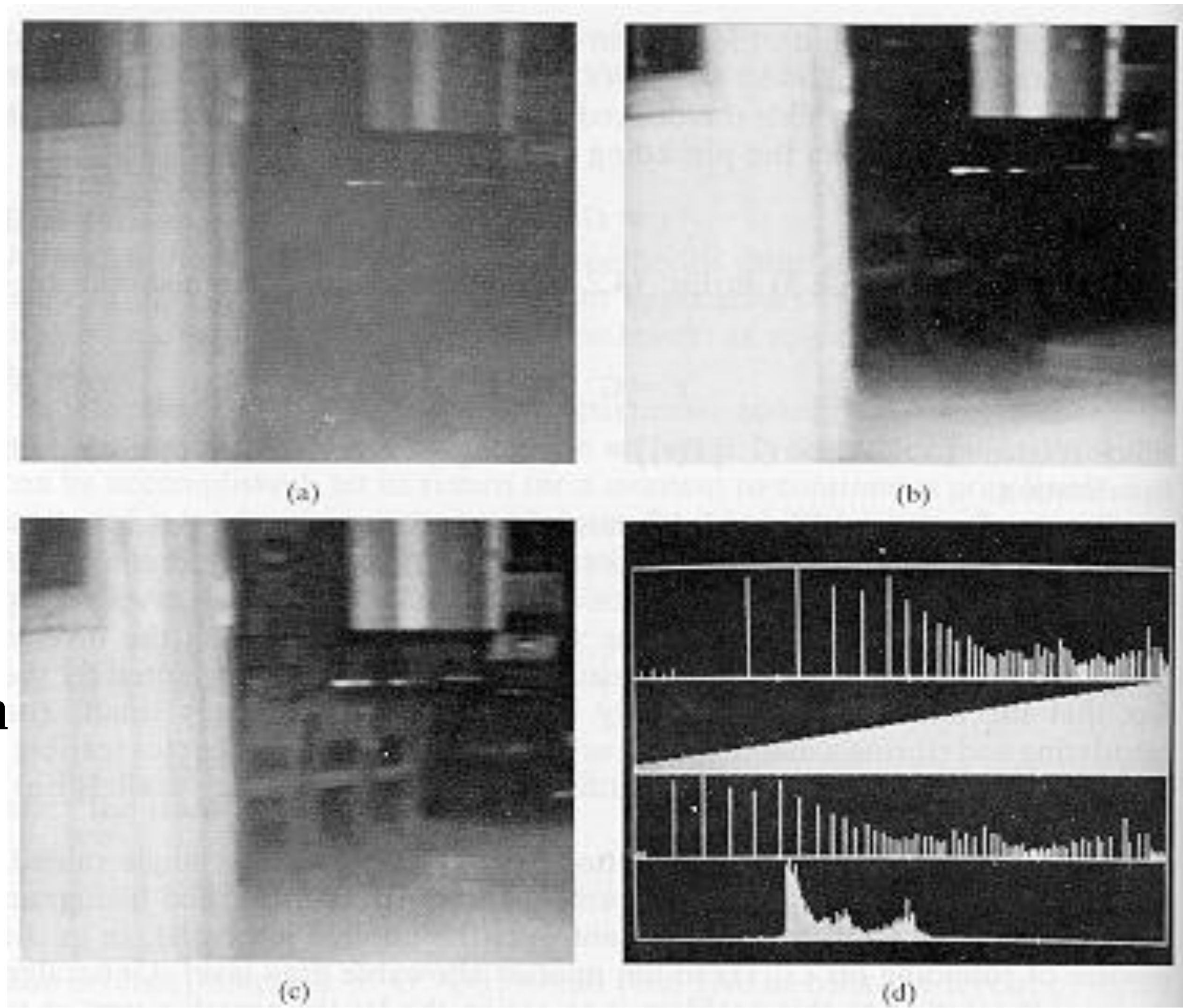
Another Example of Histogram Specification



a	b
c	d

FIGURE 3.22 (a) Histogram of a 3-bit image. (b) Specified histogram. (c) Transformation function obtained from the specified histogram. (d) Result of performing histogram specification. Compare (b) and (d).

Example of Histogram Specification



original

histogram equalization

histograms

resulting

specified

equalized

original

histogram Specifi-caction

Histogram Equalization for Color Picture

- To apply HE to an RGB color image:

  1. Convert RGB to YUV (or HSI/HSV)

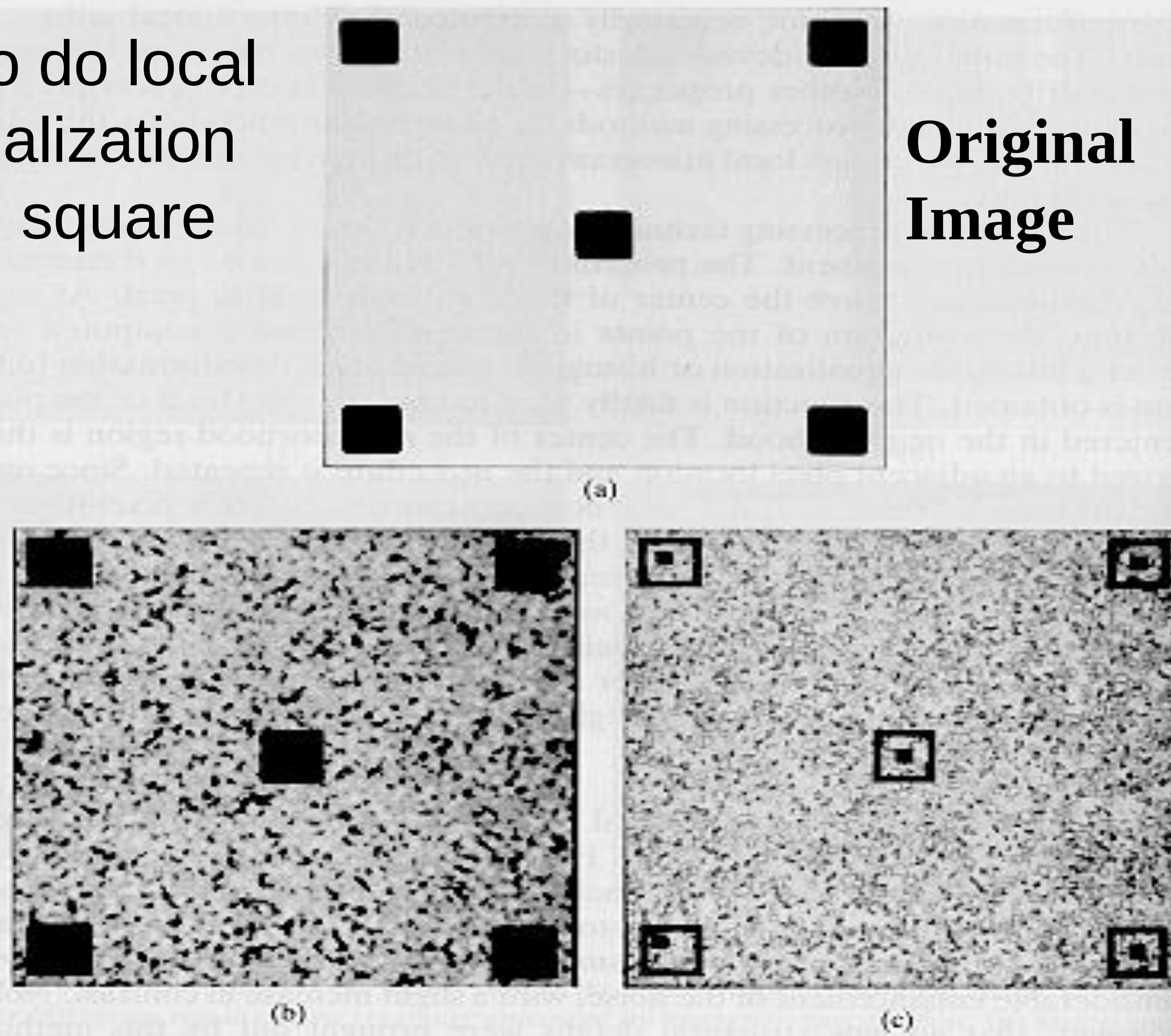
  2. Apply HE to the Y (or I/V) component

  3. Convert back to RGB

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Global vs. Local Equalization

It is possible to do local histogram equalization using a sliding square window.



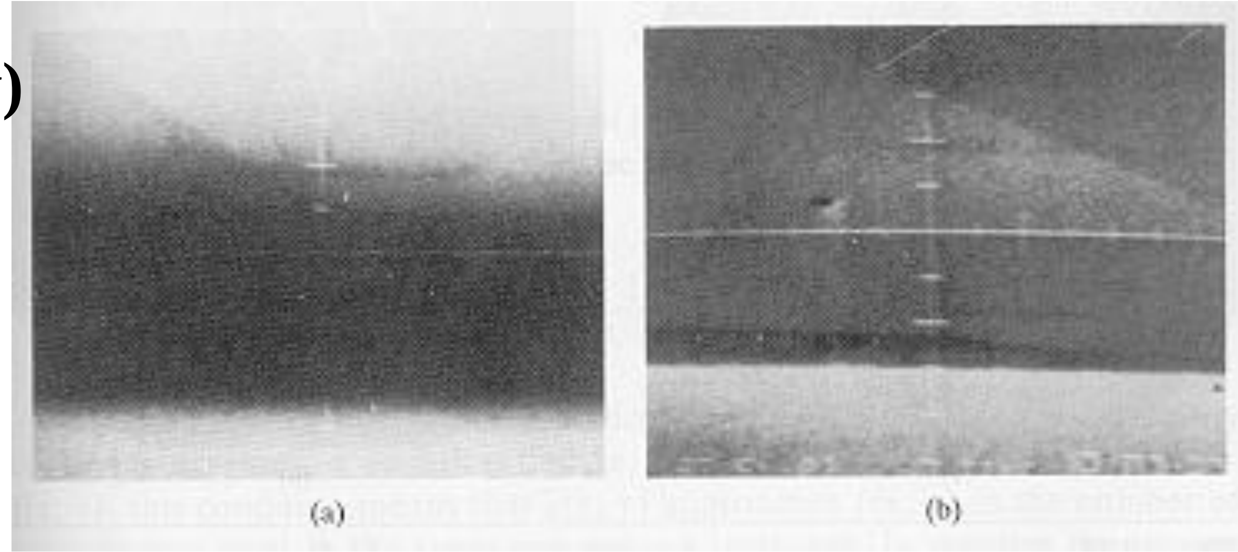
Original Image

Global HE

Local HE (7x7)

Local Enhancement

f(x,y)



g(x,y)

Using a sliding window of 15x15 pixels

$$g(x,y) = A(x,y)\left[f(x,y) - m(x,y)\right] + m(x,y)$$

$$\text{where } A(x,y) = K\frac{M}{\sigma(x,y)}, \quad 0 < K < 1$$

$$A_{\min} < A(x,y) < A_{\max}$$

$m(x,y)$: local mean in the window centered at (x, y)

$\sigma(x,y)$: local standard deviation

M: global mean

Image Averaging

- A **noisy** image $y(n_1,n_2)$ is created by adding noise $d(n_1,n_2)$ to an original image $x(n_1,n_2)$, i.e.,

$$y(n_{1,}n_2) = x(n_{1,}n_2) + d(n_{1,}n_2)$$

where *d( )* is **uncorrelated** and has **zero mean**.

- By averaging a set of noisy images, it is possible to reduce the noise effect.

$$\tilde{y}(n_1,n_2) = \frac{1}{M}\sum_{i=1}^{M} y(n_1,n_2).$$

$$E\left[\tilde{y}(n_1,n_2)\right] = x(n_1,n_2).$$

Image Averaging

$$\tilde{y}(n_1, n_2) = x(n_1, n_2) + \tilde{d}(n_1, n_2)$$

- The new noise $\tilde{d}(n_1, n_2)$ has a smaller variance (still zero mean), i.e.:

$$\sigma_{\tilde{d}}^2 = \frac{1}{M} \sigma_d^2$$

Noise Reduction by Image Averaging

- (a) noisy picture
- (b)-(f): averaging 2, 8, 16, 32, and 128 different noisy images

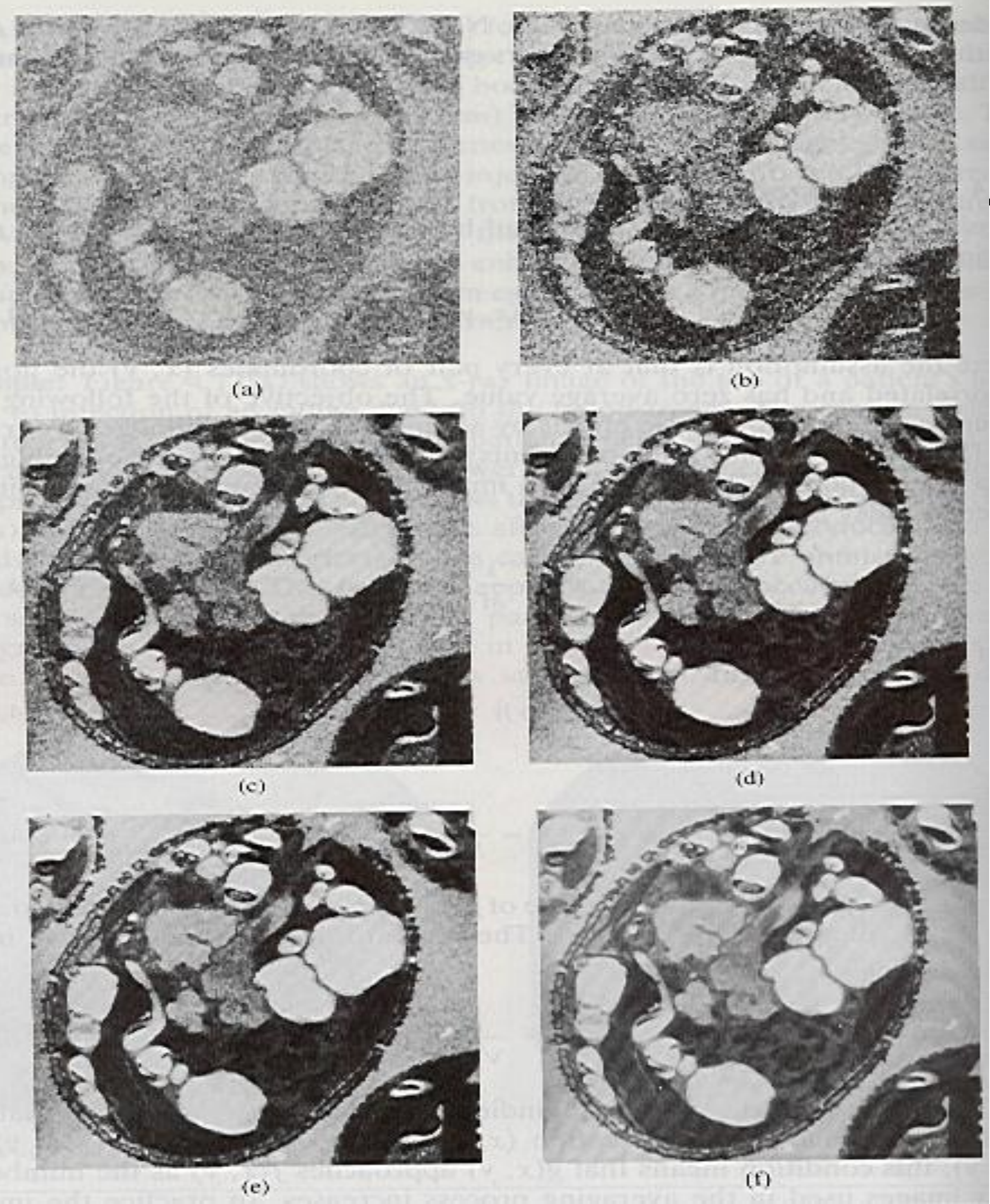


Image Manipulation

Affine Transform

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & e \\ c & d & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- **Translation, scaling, rotation, shearing, reflection, …**

Translation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & e \\ 0 & 1 & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$x' = x + e = 2$

$y' = y + f = 3$

Scaling

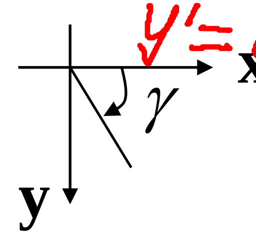
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & 0 & 0 \\ 0 & d & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$x' = ax$

$y' = dy$

Rotation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



These operators can be cascaded by matrix multiplications.